Faculty of Electronic and Computer Technology and Engineering

DEVELOPMENT OF IOT BASED SMART CLOTHESLINE SYSTEM USING RAIN DETECTION SENSOR

NUR MAISARAH BINTI MOHD KAMAL

Bachelor of Computer Engineering Technology (Computer Systems) with Honours

2024

DEVELOPMENT OF IOT BASED SMART CLOTHESLINE SYSTEM USING RAIN DETECTION SENSOR

NUR MAISARAH BINTI MOHD KAMAL

A project report submitted
in partial fulfillment of the requirements for the degree of
Bachelor of Computer Engineering Technology (Computer Systems) with Honours

Faculty of Electronic and Computer Technology and Engineering

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2024

UNIVERSITI TEKNIKAL MALAYSIA MELAKA
FAKULTI TEKNOLOGI DAN KEJURUTERAAN ELEKTRONIK DAN KOMPUTER

BORANG PENGESAHAN STATUS LAPORAN
PROJEK SARJANA MUDA II

Tajuk Projek : DEVELOPMENT OF IOT BASED SMART CLOTHESLINE SYSTEM USING RAIN DETECTION SENSOR

Sesi Pengajian : SEMESTER 1 2023/2024

Saya Nur Maisarah Binti Mohd Kamal mengaku membenarkan laporan Projek Sarjana Muda ini disimpan di Perpustakaan dengan syarat-syarat kegunaan seperti berikut:

1. Laporan adalah hakmilik Universiti Teknikal Malaysia Melaka.
2. Perpustakaan dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan dibenarkan membuat salinan laporan ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. Sila tandakan (✓):

☐	**SULIT***	(Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)
☐	**TERHAD***	(Mengandungi maklumat terhad yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan)
✓	**TIDAK TERHAD**	

Disahkan oleh:

(TANDATANGAN PENULIS)

Alamat Tetap: 5D-5-22 TINGKAT KENARI SUNGAI ARA, 11900 BAYAN LEPAS, PULAU PINANG

TS. MOHD SA'ARI BIN HAMID
Pensyarah
Fakulti Teknologi dan Kejuruteraan Elektronik dan Komputer (FTKEK)
Universiti Teknikal Malaysia Melaka (UTeM)

(COP DAN TANDATANGAN PENYELIA)

Tarikh: 17 JANUARY 2024

Tarikh: 19 JANUARY 2024

*CATATAN: Jika laporan ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa/organisasi berkenaan dengan menyatakan sekali tempoh laporan ini perlu dikelaskan sebagai SULIT atau TERHAD.

DECLARATION

I declare that this project report entitled "Development of IoT Based Smart Clothesline System Using Rain Detection Sensor" is the result of my own research except as cited in the references. The project report has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature :

Student Name : NUR MAISARAH BINTI MOHD KAMAL

Date : 17 JANUARY 2024

APPROVAL

I hereby declare that I have checked this project report and in my opinion, this project report is adequate in terms of scope and quality for the award of the degree of Bachelor of Computer Engineering Technology (Computer Systems) with Honours.

Signature :

Supervisor Name : MOHD SAAD BIN HAMID

Date : 19 JANUARY 2024

DEDICATION

To my beloved mother, Mariani Binti Abdul Samad, and father, Mohd Kamal Bin Mausen,

and

To my sisters, Nur Imanina, Nur Athirah, and Nur Qistina.

ABSTRACT

The sun-dry method is commonly used in many households in Malaysia. The sun-dried clothes method is much cheaper than buying a clothes dryer and dryer sheets. It is technically free and saves money. However, it will rain at any time without realising it in Malaysia. If residents failed to notice the rain and were slow to lift it, an almost-dry shirt on a clothes hanger outside the house might become wet again. So, the main objective is to develop a smart clothesline system to produce an alarm to help users be aware of the presence of rain from wet shirts hanging outside. This project uses an IoT-based microcontroller to detect the presence of rain. When rain drops fall on the sensor, it will create a conducting path and activate the water detector sensor. Once the water detector sensor is triggered at 30%, the motor will move. The system also controls the Firebase to store the data sensors and send them to the application that was designed. Since the system was able to detect a raindrop, it will help the housewife and the users by removing the clothes to dry places.

ABSTRAK

Kaedah pengeringan matahari biasa digunakan dalam kebanyakan isi rumah di Malaysia. Kaedah pakaian yang dijemur di bawah sinar matahari adalah jauh lebih murah daripada membeli pengering pakaian dan cadar pengering. Ia secara teknikalnya percuma dan menjimatkan wang. Bagaimanapun, hujan akan turun pada bila-bila masa tanpa disedari di Malaysia. Jika penduduk gagal menyedari hujan dan lambat mengangkatnya, baju yang hampir kering pada penyangkut baju di luar rumah mungkin basah semula. Jadi, objektif utama adalah untuk membangunkan sistem jemuran pintar untuk menghasilkan penggera bagi membantu pengguna menyedari kehadiran hujan dari baju basah yang tergantung di luar. Projek ini menggunakan mikropengawal berasaskan IoT untuk mengesan kehadiran hujan. Apabila titisan hujan jatuh pada penderia, ia akan mewujudkan laluan pengalir dan mengaktifkan penderia pengesan air. Sebaik sahaja sensor pengesan air dicetuskan pada 30%, motor akan bergerak. Sistem ini juga mengawal Firebase untuk menyimpan penderia data dan menghantarnya ke aplikasi yang telah direka bentuk. Memandangkan sistem itu dapat mengesan titisan hujan, ia akan membantu suri rumah dan pengguna dengan menanggalkan pakaian ke tempat kering.

ACKNOWLEDGEMENTS

First and foremost, I would like to express my gratitude to my supervisor, Ts. Mohd Saad Bin Hamid, his precious guidance, words of wisdom and patient throughout this project. He helped and guided me a lot when I had difficulties implementing this project.

I am also indebted to Universiti Teknikal Malaysia Melaka (UTeM) and both parents for their financial support, which enables me to accomplish the project. Not forgetting my fellow colleagues, for their willingness to share their thought and ideas on the project.

My deepest appreciation goes to my parents and family members for their love and prayers during the period of my study. My family members have given me a lot of support and also understand my efforts to make this project a success. An honourable mention also goes to Muhammad Zakwan bin Muhamad for all the motivation and understanding. He has helped me a lot in terms of giving ideas and views.

I would like to thank all the staffs at UTeM, fellow colleagues and classmates, the faculty members, as well as other individuals who are not listed here, for being co-operative and helpful.

Last but not least, I want to thank me for believing in me, I want to thank me for doing all this hard work, I want to thank me for having no days off, I want to thank me for never quitting, I want to thank me for always being a giver and trying to give more than I receive.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

mL	-	milliLitre

LIST OF ABBREVIATIONS

DC	-	Direct Current

CHAPTER 1

INTRODUCTION

1.1 Background

Without alert, it will start to rain at any moment. Most of the clothing hanging outside the home would get wet once more if we failed to notice the rain and were slow to stop it. An alarm rain detector can alert individuals when it begins to rain, allowing them to immediately change out of their clothes. When exposed to rain, the detector may pick up even the smallest amount of water and activate an alarm. A notification will be sent to the user's phone indicating that it has rained and that the clothes outside must to be lifted or changed to a more sheltered place. If the user is not at home, the system will move the clothes using a direct current (DC) motor to a more protected place. By supporting energy efficiency, improving time and resource management, lowering carbon emissions, fostering behavioural change, and addressing accessibility and affordability issues, IoT-based smart clothesline systems with rain detection sensors can also support sustainable development. The device employs low-power IoT technology and renewable energy sources to maximise energy efficiency and automatically retracts the clothesline when rain is detected. It also encourages the use of natural drying methods. Users can remotely monitor and control systems, which saves time and energy and provides real-time data to promote sustainable practises. It encourages equitable growth and contributes to a more sustainable future because it is easily accessible and reasonably priced.

Figure 1.1: Clothesline [1]

1.2 Addressing Rainfall Through Weather Sensing Project

The majority of the region's or country's yearly rainfall falls during Malaysia's rainy season, sometimes referred to as the wet season. Depending on where in the world you are, the phrase "rainy season" might have different meanings. In this case, the monsoons dominate Malaysia's wet season, which results in constant heavy rain. While in Cuba, the rainy season is characterised by heavy nighttime thunderstorms and rare but powerful tropical cyclones (typhoons). The Southwest Monsoon, which begins in late May and lasts through September, and the Northeast Monsoon, which lasts from November to March, are the two monsoon seasons that characterise Malaysia's weather. While the Southwest Monsoon often shows drier weather, the Northeast Monsoon brings significant rain, particularly to Malaysian states on the eastern coast of Peninsular Malaysia, west Sarawak, and east Sabah. The inter-monsoon phase is the period of time between these two monsoons. DHT22 sensors are widely used to measure temperature and humidity [1], and in the past, they were combined with other microcontrollers, including the ESP32, to create weather stations.

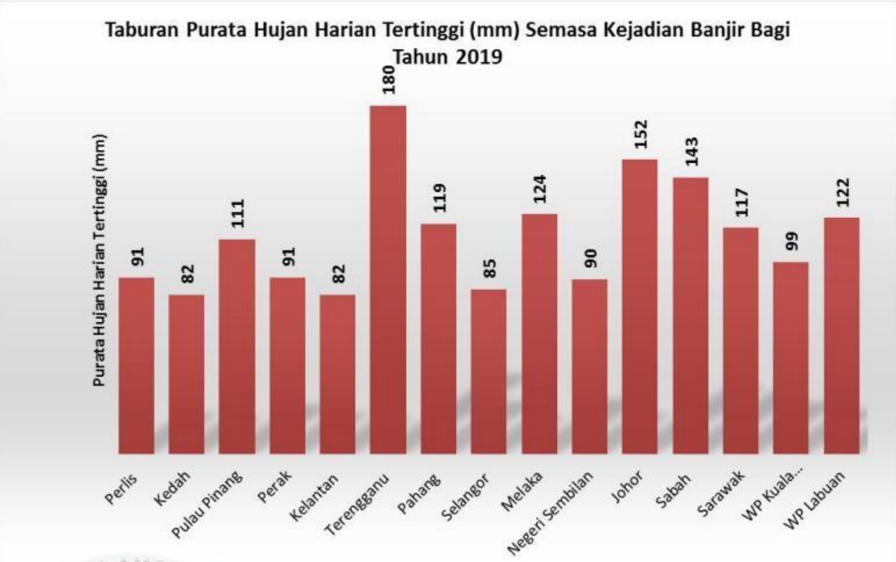


Figure 1.2: Average Rainfall in Malaysia on 2019 [3]

1.3 Problem Statement

Even though users have other commitments. As an example, a housewife who is busy with cooking or other tasks at home or an employee who has already gone to work that day may not realize that it has rained. The clothes on the clothesline will get wet and it will take time to dry the clothes again. We can justify that rain sensor users are very important for this project based on their use in each research in chapter 2. However, from the current system will be developed once more by developing the application interface rain detection system using rain, humidity, and temperature sensor to produce a right judgement, and in the development of the side of its rule basis is renewed once more [2]. It also can affect our daily life.

1.4 Project Objective

The aim of this project is to propose a systematic and effective methodology to optimizes the clothes drying process. Specifically, the objectives are as follows:

a) To perform literature study of clothesline system using rain detection sensor.

b) To develop the prototype of the clothesline system using rain detection sensor.

c) To analyze the accuracy the realibility system of the clothesline system using rain detection sensor.

1.5 Scope of Project

The scope of this project are as follows:

a) Determine and select a suitable rain sensor that can detect raindrops with accuracy. Get real-time information about rainfall intensity or presence by integrating the rain detecting sensor with the NodeMCU ESP32 microcontroller.

b) Identify the smart clothesline system's overall architecture and design.

c) The voltage used for the DC motor is only 5V, to control the movement of the clothesline based on the rain detection.

d) Design a user-friendly interface such as Android Studio on mobile app, to let users check on the state of the system, and get notifications/alerts about rain detection and clothesline actions.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

In the modern day, the introduction of IoT technology has transformed every aspect of our life by providing creative solutions and encouraging productivity. The development of an IoT-based smart clothesline system which includes rain detecting sensors is the main topic of this literature study. Traditional clothesline drying is heavily dependent on the weather, which results in uncomfortable moist clothes. This intelligent system can detect rain precisely and decide how best to dry clothing by integrating a rain detection sensor. The evaluation examines technological elements, design factors, implementation issues, and prospective advantages including energy savings and better clothing care. In order to advance the subject and encourage sustainable living, it also reviews pertinent research in smart home systems, IoT applications and sensor technologies.

2.2 Internet of Things (IoT)

The Internet of Things (IoT) explains the network between physical objects that are embedded with sensors, software and other technologies for connecting and exchanging the data with other devices and other systems over the internet. The article discusses the development of a prototype for an automatic clothesline system. The system includes various sensors, including light sensors, rain sensors and humidity sensors to automate the drying process based on the environmental conditions. The prototype also highlights the use of IoT to improve the effectiveness of daily routines by providing options for both manual control and automatic operation via a smartphone interface [5]. Same goes to another article, an inventive solution is put up in the shape of an Internet of Things-enabled Automatic Clothesline Retrieval Prototype with a Humidity Alert System in response to the growing difficulties people encounter when handling laundry in an environment of hectic schedules and unexpected weather. Acknowledging the limitations created by users being away from home and the challenge of forecasting rain, the system makes use of Internet of Things technologies to offer automation and real-time monitoring [8].

Using Internet of Things technology, the Rain Monitoring and Prediction System described in [10] addresses how rainfall affects natural disasters like floods, landslides, and droughts. The system uses IoT sensors to gather data on rainfall patterns and intensities, with an emphasis on the northeast monsoon's impact on Peninsular Malaysia's east coast. The project emphasises the integration of hardware and software components through its architecture, which includes circuit schematics, user interfaces, a database, an IoT system architecture, and a mobile application environment. The usage of water level sensors, rain sensors, and GSM SIM800C Modules, which enable real-time data transmission to a cloud platform and a mobile application. Through testing in different environments, the system's efficiency has been evaluated, indicating its potential to improve rain monitoring skills.

The Automatic Rain Sensing Car Wiper project, as detailed in [11], presents an Internet of Things (IoT)-based method to improve car safety in rainy weather. The technology makes use of a rain sensor that communicates with an Internet of Things-based control board when it detects water on the windscreen. The ESP8266 module powers this control board, which controls the automatic on and off of the windscreen wipers. The integration of wireless communication and data transmission to the car's Body Control Module (BCM) highlights the Internet of Things component. The block diagram displays the signal flow and demonstrates how IoT technology helps to improve wiper efficiency by giving real-time data. The project's design highlights the benefits of integrating IoT into automotive applications by ensuring low power consumption, dependability, and the capacity to operate either manually or automatically.

Using IoT technology and ESP32 microcontroller, the Solar Water Pumping System Control project described in [12] provides underdeveloped countries with an automated and reasonably priced irrigation system. In order to gather information for effective water management, the system makes use of temperature, humidity, and soil moisture sensors that are connected to the ESP32. Through the use of a web server, users can remotely control the irrigation system, with the ESP32 serving as a central control unit that makes decisions depending on data received. The project demonstrates the benefits of utilising the low-cost, low-power, and integrated connectivity properties of the ESP32 microcontroller for Internet of Things applications, making it appropriate for agricultural automation. Farmers can monitor and manage irrigation processes thanks to the incorporation of IoT, which lowers water waste and boosts overall productivity.

2.3 Weather Monitoring

The implementation of weather monitoring into IoT-based devices introduces a degree of intelligence and flexibility. Many applications depend heavily on weather conditions, and adding real-time weather data can improve systems' efficiency and effectiveness. This subtopic examines the value of incorporating weather monitoring into Internet of Things systems, emphasising the two projects that were previously covered. The rain detection system that was previously discussed which is Rain Detection System For Estimate Weather Level Using Mamdani Fuzzy Inference System [4] is dependent on sensors, such as the DHT 22 for temperature and humidity and a rain sensor for rainwater detection. Integrating weather monitoring capabilities would entail utilising third-party weather APIs or modules that offer more comprehensive regional weather data in order to increase accuracy and responsiveness. This improvement may result in more reliable rain prediction decisions, enabling the system to plan for rain based on more extensive weather patterns.

In order to provide an automatic drying system, sensors are used in the Internet of Things-based clothesline and air drier prototype [5] to measure light, rain, and humidity. To build on this, the system might be integrated with systems that provide weather forecasts in order to predict future weather patterns. This would make it possible for the system to make wise choices, such delaying drying operations when rain is expected to avoid clothes becoming wet again. The success and efficiency of some automated systems are significantly influenced by the weather. For example, the capacity to detect rain is important for keeping clothing dry in the Automatic Clothesline Puller [7] and the Automatic Clothesline Retrieval Prototype [8]. In addition to rainfall, weather monitoring can provide other data like temperature, wind speed, and humidity, all of which can improve the efficiency of these systems even more. By using sensors that can measure a variety of meteorological factors, weather monitoring can be implemented. Weather sensors can be added to the ESP32 microcontroller, which is currently a key element of the Home Automation System [6]. The real-time database Firebase, which powers the home automation system, can receive data from these sensors, enabling cloud-based storage and analysis.

By using sensors that can measure a variety of meteorological factors, weather monitoring can be implemented. Weather sensors can be added to the ESP32 microcontroller, which is currently a key element of the Home Automation System [6]. The real-time database Firebase, which powers the home automation system, can receive data from these sensors, enabling cloud-based storage and analysis. By including a humidity sensor, the Automatic Clothesline Retrieval Prototype [8] can gain from weather monitoring in the context of laundry management. The technology may notify customers when ideal drying conditions are met or alert them to sudden weather changes that could impact the drying process by analysing humidity levels.

2.4 Previous Related Works

2.4.1 Rain Detection System for Estimate Weather Level Using Mamdani Fuzzy Inference System [4]

Many investigations have looked into developing a system for detecting rain using a neural network and genetic algorithm. However, from the current system will be developed once more by developing the application interface rain detection system using Rain, Humidity, and Temperature sensor to produce a right reasoning, and in the development of the side of its rule basis is renewed once more. The open-source Arduino Microcontroller is a single-board microcontroller. The purpose of Arduino is to simplify things for individuals that work with electronics. A common boot-loader programming language compiler makes up the Arduino software. C is the chosen programming language. Several libraries that are helpful to support the complexity of hardware and software integration have been installed on an Arduino microcontroller. A digital sensor called the DHT 22 is used in measuring the humidity and temperature of the air around it. When the internal sensor detects something, the module will immediately offer a temperature measurement that corresponds to its surroundings because calibration coefficients are stored in the OTP programme memory. Specifications for humidity sensors which are supply voltage (5V), range of temperature measurements from -40 to 80-degree celcius, humidity measurement range from 0 − 100% RH, the size of humidity sensor is 15.1 mm × 25 mm × 7.7 mm. One of the sensors that is sensitive to rainwater is the rain sensor. The operation of this rain sensor involves connecting a ground path and a port path so that, when the sensor is exposed to rainwater, the port is directly connected to the ground and there is no voltage. At the level of rainwater electrolysis, the rain sensor delivers input value. Problems that uncertainty and a lack of mathematical model control processes contain are represented using fuzzy logic. Based on the user experience of the control system, fuzzy logic offers a technique of formula for representation, manipulation, and implementation. The membership function or membership function is crucial in suppressing the issue and generating an appropriate conclusion. The three basic parts of a fuzzy rule-based system are fuzzification, inference, and defuzzification. Two different kinds of sensors provide data for this system to process. First, the air discharge sensor for every time it rains. After that, a humidity sensor for temperature was used. The fuzzy logic method is used by this system to offer a perception of rainfall. The three sensors' results were used to generate the rules based on actual conditions. The analysis of a rain detection system employing a single board of temperature, humidity, and rain sensors that can produce trustworthy rule conditions using Mamdani FIS is a contribution made by this system's research. Arduino Uno, Sensor (Rain Sensor, Humidity, Temperature), Fuzzy Control Interfacing Visual Studio 2006, and Decision Making Using a Fuzzy Logic Method, which is One of the Mathematical Method Models, are the major parts of this system. The use of information technology is expanding quickly today. Using a neural network and genetic algorithm, there are still some strategies that can be created once more in the fields of weather forecast prediction, various human activities such as agriculture, water resources, hydroelectric power projects, occurring droughts or floods and others.

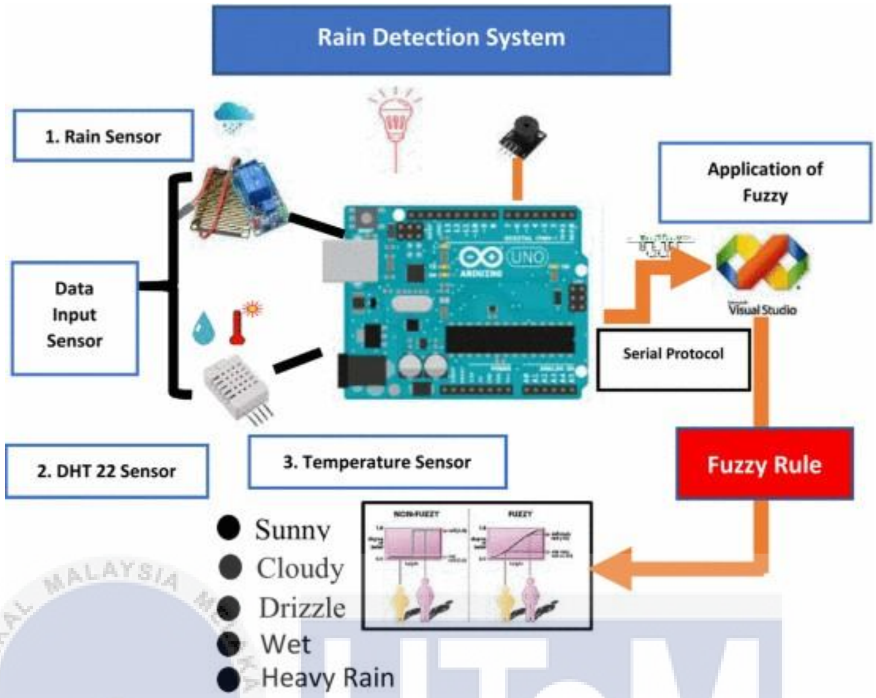


Figure 2.1: Rain Detector System Architecture [4]

2.4.2 Design and Implementation of Clothesline and Air Dryer Prototype Base on Internet of Things

Based on the previous article titled "Design and Implementation of Clothesline and Air Dryer Prototype Base on Internet of Things [5]", if you must wait and keep the clothing dry, drying garments takes a lot of time. The rain frequently causes previously dried clothing to become wet once more, and for a variety of reasons, nobody picks them up. In addition, long-term exposure to the elements, particularly high humidity levels, can result in the growth of fungus on clothing. The process starts with a literature review, problem identification, and needs analysis. Next, a system is designed, with inputs from several sensors, including light sensors that measure brightness with luxmeters, rain sensors that detect the presence of rain, and humidity sensors that measure the humidity in the air near the clothesline. To make a prototype that is simple to implement, the sensors are merged. Both system automation and its disabling are options. The system will operate based on the input data the sensor receives if the automatic mode is activated. When the automatic mode is disabled, the system will operate in accordance with the user's smartphone commands. Tools must have a functional system design to perform as intended. The earliest input to the last output are covered by a workflow of tools in this work system design. The internet-based automatic clothesline and dryer prototype's system/workflow design is presented. The program's interface design is used by smartphone users as a user interface display so they can keep an eye on and manage the tool (in this case, a clothesline and an automatic dryer). The smartphone's user interface in view. For the purpose of testing rain sensors, water is drip-fed onto the sensor using a pipette to simulate rain. The experiment involved changing the quantity of water droplets used (it is known that each pipette drop has a volume of 0.05 mL). If the indicator light illuminates when there is water on the sensor and turns off when dry, the sensor is accurate and worthwhile utilising. When automated mode is turned on, the tool will operate in accordance with the data that the sensors collect. This test seeks to determine the system reaction delay as well as the correctness of the tool's system response to different input variations. Using a stopwatch, the delay time is determined starting when the auto mode button is depressed and ending when the actuator (motor and fan) moves. The outcomes of automated mode testing can be seen with the fan turned off and the clothesline (motor) in its initial position outside. Modern human life is extremely active and competitive. Most people nowadays, both sexes, prioritise their careers over everyday responsibilities, so minor chores like drying clothes, which can take a long time if you have to wait for the entire process to be finished, are frequently neglected. Drying clothes can be a hassle, especially now that the rainy season has arrived. Therefore, technology continues to advance and present as a solution to any difficulty that develops in human life, including the instances of problems that have been stated.

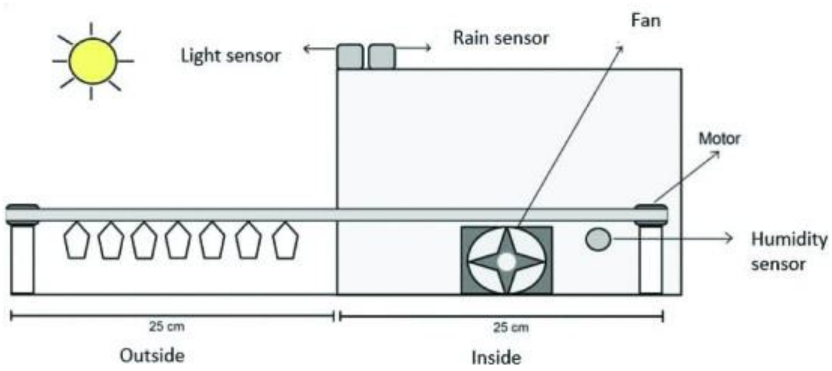


Figure 2.2: Project design for Design and Implementation of Clothesline And Air Dryer Prototype Base on Internet of Things [5]

2.4.3 Home Automation System Using ESP32 and Firebase [6]

One type of technology that can regulate our home and keep us linked to it is house automation. The term "home automation" describes the interconnectedness of all appliances and home goods. For instance, a central microcontroller panel would be able to handle anything from heating to standard electrical equipment. With home automation, we may remotely control our house's features using a computer or any mobile device. This allows us to programme electronic equipment based on scenarios or conditions, or to centralise control of a few gadgets in one location. Communication between multiple controlled devices is essential. Home automation's main objective is to control or keep an eye on signals coming from various devices or important services. The home automation system may be controlled and monitored by a smartphone. The ESP32 microcontroller, which is used to build the home automation, and Firebase, a real-time database used in this study to store data on the cloud, are used. Additionally, Next JS is used to develop and host a webpage for the virtual switches on Vercel. With this, four relays may be remotely controlled through the internet, allowing users to see feedback in real time. We now have access to high-tech functionality and elegance that was previously beyond of our reach thanks to smart home automation. With home automation, almost all household appliances may be managed remotely and frequently via a single app. Home automation gives us a lot of options, including the ability to tailor some elements to our preferences and so improve convenience. This project intends to keep manual controls in case there is no internet connection while simultaneously enabling remote access to our household appliances.

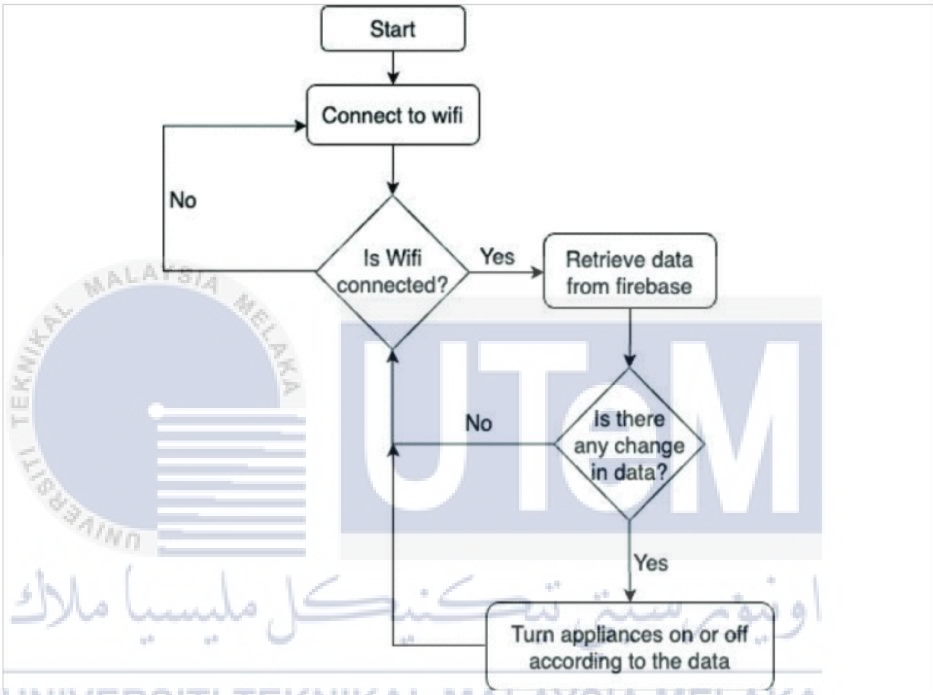


Figure 2.3: Flowchart of Home Automation System Using ESP32 and Firebase [6]

2.4.4 Automatic Clothesline Puller [7]

For the purpose of assisting individuals with their laundry issues, the automatic Clothesline Puller was created. The people responsible for laundry work may feel less burdened as a result of this effort. This project is appropriate for home gadgets that can lessen a housewife's workload. When we are at home, we tend to notice trivial things like drying clothing outside the house. When a raindrop is spotted, the proposed project will function by causing everyone to get their garments and go inside. The project's primary operating premise is to utilise a rain detector to detect raindrops and then activate a device to draw laundry from the clothesline inside the house. An alert or buzzer is automatically activated when rain (rainwater) is detected, which is a straightforward but useful project. Everyone's life depends on water in some way. Water conservation and wise water use are crucial. One of the market samples available. The rain sensor that is typically seen on the windscreen of a car may be installed in some market vehicles, particularly autos. Most of the time, we miss things because we think they are just a stripe on the back of the rearview mirror. In order to prevent mist from forming on the back shield, this sensor on the automobile can help turn on the wiper, help turn on the heater for the back shield, and help keep the rear glass warm. As technology advances along the IR 4.0, the globe has changed many paradigms and dimensions today. We are willing to use this enhancement and benefit from those technologies as one of these users. Not to mention specially designed indoor and outdoor appliances for household use, both play critical roles. The term "clothesline" was first used in a sentence in 1830. Since then, until an IoT comes along, the evolution of clotheslines has been carried out using a physical motor and a manual switching method.

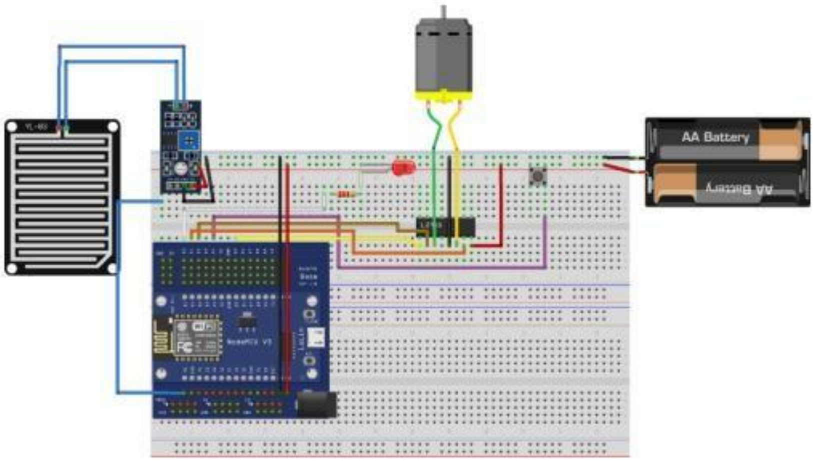


Figure 2.4: Proposed Project Circuit of Automatic Clothesline Puller [7]

2.4.5 Automatic Clothesline Retrieval Prototype with Humidity Alert System to Aid Clothesline Drawbacks for Reducing Laundry Worries [8]

Nowadays, people have a lot of work on their plates both as adults and as students. Laundry drying on the clothesline is one of the most troublesome tasks in the home. The user's busy schedule from classes and mounting work prevents them from checking on their garments hanging out to dry when a heavy downpour occurs, resulting in soggy clothing. In addition, individuals find it challenging to keep their garments dry and find it tough to lift their clothes when it rains because of the unpredictable weather. Since users won't always be at their homes, it is difficult to predict rain in the current situation. When it rains, individuals who are present can take the garments themselves to a dry location. Even when users are present in their homes, it takes some time for them to recognise the impending rain and take the necessary precautions, such as putting their clothes somewhere safe. The garments will become wet when it rains if they are not at home. If there were an automated system that could help individuals manage their laundry in their daily lives, that would be convenient. The gathering and analysis of requirements constitutes the first stage. Using the prototype model as a reference, this step involves creating a time planner for the project using a Gantt Chart in Microsoft, conducting a survey, and analysing the relative humidity of the surrounding environment. The quick design phase, which comes after the first, will focus on creating prototype sketches. A technical sketch depicts the interaction between each component of the prototype while a physical sketch depicts the prototype's overall appearance. The third phase will be the prototype building process phase, which will detail the system's and the prototype's construction procedure step-by-step. The Arduino IDE and Visual Studio will be used to programme the system's components and application. The initial user evaluation is the next phase, which is the fourth overall. To determine how well the prototype and the application function, a test will be conducted on the finished prototype model. The prototype refine phase, which is the second to final phase, is where the issues that were discovered in the first phase will be fixed. The preceding stage will be performed when all issues have been resolved to check for new issues. The system implementation and maintenance product is the final phase. In this phase, the effectiveness of the system will be observed, and all issues found will be resolved. For the Humidity sensor, the atmosphere is dry, cloudy and rainy are tested with cotton facial pads tissue to mimic such situations. The user's daily life will become more efficient thanks to this system, and they won't have to worry about anything, especially their washing. The simple alarm system included with this project can reassure the user about the status of their laundry and clothesline. If the user does not want to get repeated alerts about the condition, the system can be improved by lowering or restricting the amount of alerts that will be delivered to the user. A better motor can support a clothesline that can manage a greater weight.

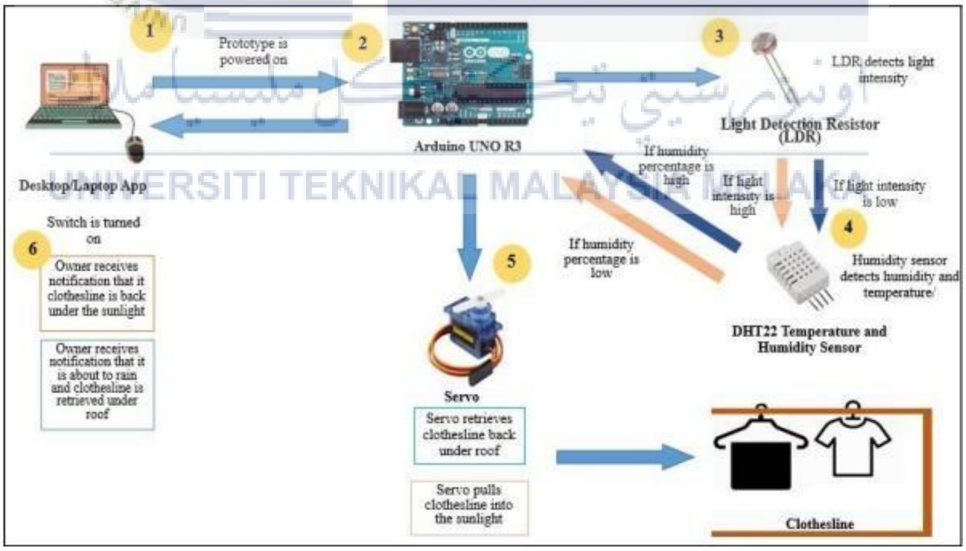
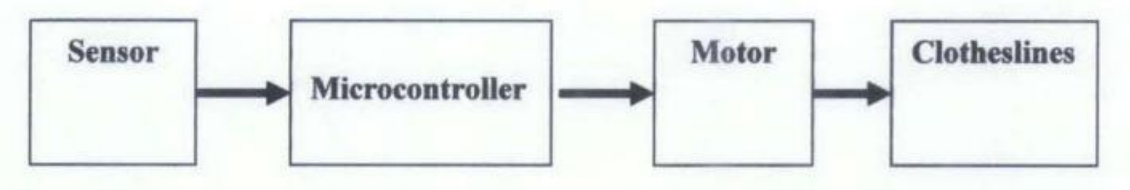


Figure 2.5: The Physical Prototype Sketch [8]

2.4.6 Automated Clothesline System

This previous chapter titled “Automated Clothesline System [9]” that the users are not always at their homes, it is difficult to predict rain in the current situation. When it rains, individuals who are around can take their clothes to a dry location. Even when users are actually in their homes, it takes some time for them to notice the coming rain and take the necessary precautions, such as putting their clothes somewhere safe. Their clothes will become wet when it rains if they are not at home. If there were an automated system that could help individuals manage their laundry in their daily lives, that would be convenient. Consequently, an automated system that can recognise incoming rain and bring their clothes to the safe place is needed so that daily routines can be done easier. To find the problem solutions, the project's problem statement was thoughtfully written. The most effective strategies to carry out the project were then determined through research and studies. Studies and research on materials were done, including what hardware and software would be best for the project as well as the cost of the supplies. The ultimate design was decided upon during the investigations that were carried out. For the implementation of the project, several tools were required to be used to attain the goals. For hardware, we use water level sensor circuit, PIC microcontroller, pulley and string for replaced by track and gear, 12V DC motor. For software, we apply PSpice software to draw and simulate the designed circuit, PIC C software for programming the PIC Microcontroller, WARP 13 is the burner software to burn the PIC microcontroller, eagle layout editor for schematic drawings and Gerber files creation to design the printed circuit board(PCB). Therefore, the system may either put the clothes to the safe place while it is raining or put the clothes to the unsheltered region to dry the clothes back, it is recommended that the motor can be operated in two ways (forward-reverse operations). For this need, additional sensor (such as a heat or light sensor) is required. In order to halt the motor at the appropriate time and location, it was also advised that limit switches be employed in place of the time delay function.



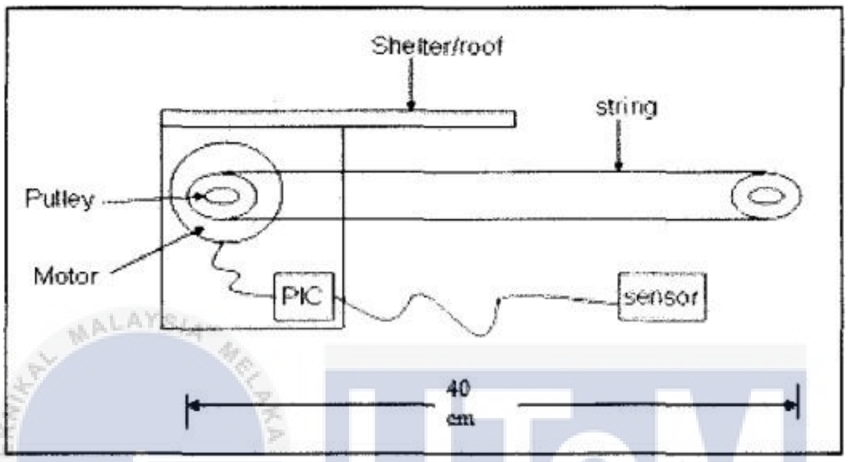


Figure 2.6: The design concept of Automated Clothesline System [9]

2.4.7 Rain Monitoring and Prediction System Using IoT [10]

Rainfall played a significant role in causing natural disasters such floods, landslides, and droughts. The northeast monsoon typically directly affects Peninsular Malaysia's east coast, which receives 2500 – 3000mm of rain annually. In November and December, there will be more than 500 mm of rain every month. The biggest amount of rain is often recorded in the month of December. Particularly in the central and southern regions, this period might persist into January and result in flooding. The data included in the researcher's requirement analysis came from journals, publications, and conference proceedings. Researchers collect and analyse requirements from many stakeholders. The researcher also examined the planned tools and hardware. In the design phase, researchers collaborate with key players to create models and prototypes that incorporate hardware, software, and system inputs and outputs. An architecture outline design that comprises designing the mobile application environment, the IoT system architecture and circuit schematic, the user interface for the mobile application, and the database is produced after the requirements acquired during the requirement analysis are analysed. Create a prototype. The development phase makes up the third stage of the prototyping process. Based on data gathered during the requirement gathering and design phases, the prototype was built. The developers process the data and combine the sensors and microcontrollers with the model during this stage. This stage seeks to improve the with the aid of a microcontroller, a reliable connection between the cloud and the microcontroller, as well as between the cloud and the mobile application, sensors can read data accurately. The main software chosen is the open-source Arduino IDE. Kodular, an online drag and drop Android app maker, was chosen as the primary software for developing the mobile application. The prototype was tested to ensure that each component is operating properly before user evaluation. To evaluate the effectiveness of the model's components, it was tested under identical conditions using a separate sensor. Rain sensors will be tested in rainy weather and installed in open spaces where it is practical to collect rainfall data. In order to collect information on higher rainfall levels, water level sensors will be installed in neighbouring rivers or other readily inundated locations. When the water level sensor detects a high amount of rainwater that exceeds the critical level, the sensor alarm will be tested. The Common Protocol for Sensor Testing was used as the testing approach. The prototype has been improved after testing. The accuracy of sensor readings that will appear on a mobile application and a cloud platform was tested in the meantime using a mobile application. Thirty respondents were given questions to complete as part of the testing of the mobile application. The product is refined at this step to make sure the errors and limitation are fixed. The sensors will be examined to see whether there is a failed connection during this step, and the source code will be checked once again. The maintenance phase comes last. The finished prototype was made public at this point in order to get user feedback for upcoming upgrades. This project made use of the KG004 rain sensor module. The low-cost sensor generates electrical differences onboard using a set of exposed parallel lines when drops or water volume changes. The FC-37 sensor's function is to find nearby rainfall. On it, there are four pins. The four (D0) are voltage (VCC), ground (GRD), analogue pin (A0), and digital pin (D0). It measures moisture with an analogue pin and shows the data with a digital pin. GSM SIM800C Module is a piece of equipment that automatically transmits and receives wireless data. It uses a 4G SIM card and operates in the 850MHz, 900MHz, 1800MHz, and 1900MHz frequency bands to deliver SMS and voice calls. Users can use it to store GSM mobile phone data. A piece of hardware called the Arduino UNO R3 is used to offer an IoT platform. The sensors and modules have all been connected. It is used to transmit and receive values retrieved from the sensors. It has a 16Mhz clock speed, a 7 to 12V input voltage range, 32k Flash Memory, 14 digital input-output pins, and six analogue inputs. A buzzer is a device that emits an alarm in response to input from an electrical device. It automatically notifies you when the readings hit the threshold level. A cheap standalone wireless transceiver that can be used in Internet of Things applications is the ESP8266 module. A specified Baud rate is used to link the ESP8266-01 module to the microcontroller via UART. A water level detector is used to measure rainfall or water levels. When water is detected, the circuits on the printed leads on its sensor board are finished. The rain sensor cannot identify a raindrop moving at less than 0.5mm/h, therefore "It is not raining" will be shown. However, the rain sensor will identify the raindrop and display "It's raining" once the raindrop speed reaches 10.0mm/h or above. At some point, the information will be transmitted to ThingSpeak, which will subsequently send the notification to the user's mobile app. More than 85% of those surveyed about the smartphone application concur that using it makes it easier to track and forecast rain and water levels. The development of science and technology in predicting atmospheric conditions at certain locations and periods is known as weather forecasting. It can be used to identify rain patterns, offer suggestions, and identify potential problems so that we can make wise judgements. There are three forms of precipitation that are tracked and measured. Drizzle rain, which has a nozzle-like diameter of about 0.5 mm, is the first type. Second, heavy rain has a temperature of more than 100 degrees Celsius and a diameter of at least 7 millimetres. Thirdly, monsoonal rainfall occurs when a band of easterly winds forms at latitudes between 30 degrees north and south due to the interaction of the Earth's rotation and heat from the sun.

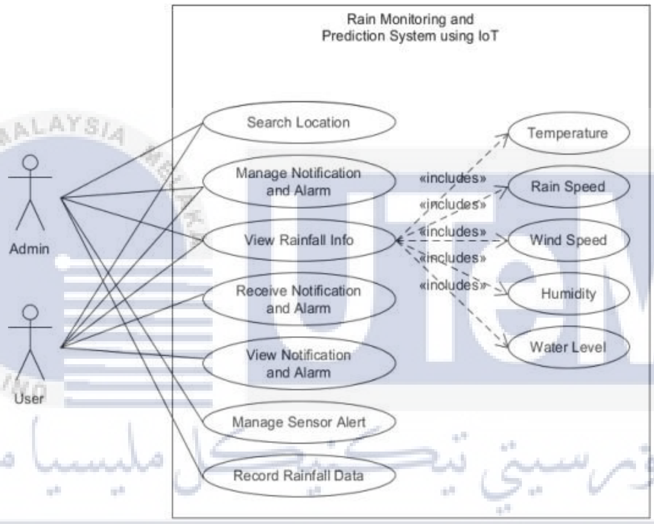


Figure 2.7: Use Case Diagram of Rain Monitoring and Prediction System Using IoT [10]

2.4.8 Design and Implementation of Automatic Rain Sensing Car Wiper [11]

We observe the act of water being sprinkled on our wheeler's front window on rainy days. Drivers of an automobile is unable to see other vehicles on the road. He then tries using the wiper on the glass, which requires him to frequently turn on, leading to a vehicle accident. If we put any kind of sensor to the glass that can detect when water is being sprayed on it, the wiper will start running on its own. Although this proposed wiper system will remove water from the windscreen, it will also be updated to an automatic control system by employing a controller. The principle of this proposed wiper system is same to that of other conventional wipers. As soon as water touches the sensor, a signal is sent to the system, driving the wiper motor. The wiper will stop after the sensor has not detected any water. This will lessen the disadvantages that were mentioned at the outset. The wiper will automatically push up from the windscreen when the motor is turned off, according to an additional invention proposal. The automatic rain sensor block diagram can be broken down into a number of blocks, including the AC supply block, which is made up of a 230V AC, the transformer block, which steps down the 230V AC to 12V AC, the rectifier and filter block, which converts the 12V AC to 12V DC and then smooths out the pulsating DC voltage, the control board block, which receives signals from the rain sensor and sends them to the DC motor, and the sensor block, which detects. The AC supply feeds the system with 230V AC which is further stepped down to 12V AC by the transformer. Transformers are electrical devices that isolate the power flow from one winding to the other winding. All electronic devices operate with lower voltages. In order to scale down the AC voltage from 230V to the necessary 12 V, a step-down transformer is utilised. The diodes are connected to the transformer's 12V AC output, which is used to rectify the signal. Infrared light is shone at a 45-degree angle into the windscreen from the interior; if the glass is wet, less light returns to the sensor and the wipers activate. This is how the majority of modern rain sensors work. This is used to regulate the wiper's speed and direction. Every time the control circuit sends a signal to run the wiper, it activates. The sensor unit mounts undetected to the inside of the windscreen and offers a larger detection area, more precision, and a more affordable price than the optical device that was previously used. As it mounts in the same spot on the inside of the windscreen and transmits the same control signals to the car's BCM, this sensor was made to be able to readily replace optical units. This design is a highly helpful and advantageous working model due to its low power consumption, lack of wear and tear, ability to function manually or automatically via an on/off switch, and other factors.

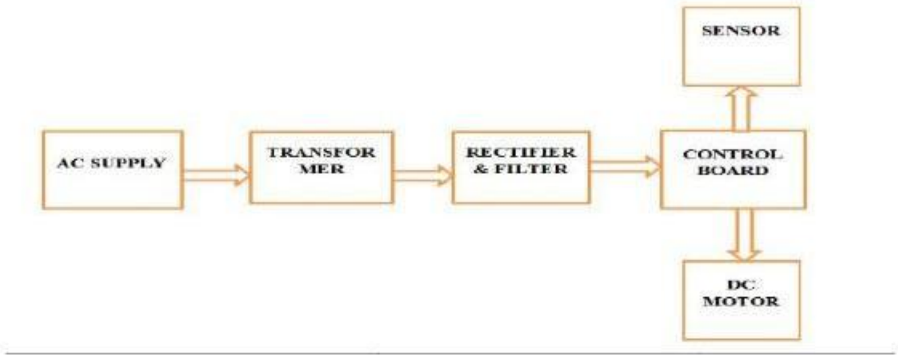


Figure 2.8: Block Diagram of an Automatic Wiper [11]

2.4.9 Solar Water Pumping System Control Using a Low Cost ESP32 Microcontroller [12]

It is an automated solar water pumping system with a low cost for irrigation in developing nations. The temperature, humidity, and soil moisture level are all detected by the programmable sensor module, which then transmits the data to the ESP32 microcontroller. Additionally, a water level sensor tracks the water level and transmits the information to the microcontroller. The microprocessor chooses whether to start or stop the pump motor based on the data and boundary conditions. It also explains how to choose the appropriate soil moisture limitations for a certain type of soil. For the user to see the findings, the ESP32 microcontroller also transmits them to the web server. With just a click on a mobile, the user can control the irrigation system from a distance. Designing and demonstrating a controlled solar water pumping system for Bangladeshi agriculture is the main goal of this project. A solar PV array, inverter, motor-pump set, and storage system make up a solar irrigation pumping system. There are two different kinds of storage systems: ones that store water in a big tank and energy in batteries. The second option will be used in this paper. HOMER (Hybrid Optimisation and Multiple Energy Recourses) was used to size the system. Designing and demonstrating a controlled solar water pumping system for Bangladeshi agriculture is the main goal of this project. A solar PV array, inverter, motor-pump set, and storage system make up a solar irrigation pumping system. There are two different kinds of storage systems: ones that store water in a big tank and energy in batteries. The second option will be used in this paper. HOMER (Hybrid Optimisation and Multiple Energy Recourses) was used to size the system. Additionally, he has the ability to manage the entire irrigation system while seated at home or away from the field. Microcontrollers and programmable logic circuits (PLC) can be used to regulate water pumping systems. These fixtures cost hundreds of dollars and have maintenance problems in emerging and impoverished nations. Espressif System unveiled the ESP32, which succeeds the ESP8266 microcontroller [14]. It is a system on a chip microcontroller that is low-cost, low-power, and has built-in Wi-Fi, dual-mode Bluetooth, and power-saving features, making it more adaptable. Applications for the Internet of Things (IoT) and mobile devices are both compatible with the ESP32. Due to its broad operating temperature range, it proved to be a dependable choice in an industrial setting. This microcontroller can be used as a fully functional standalone system or as a slave to another microcontroller. A sensor for measuring temperature, humidity, and soil moisture is the HiGrow sensor module. It can upload sensor data via cloud communication so that users can check field conditions using any internet browser. When a client connects to it via the HTTP (Hypertext Transfer Protocol), it can also be connected to the web server, which is in charge of returning a web page to the client. The HiGrow plant monitor is connected to a sensor module DHT11 for temperature and humidity. The DHT11 and ESP32 are linked such that pin 22 on the ESP32 can read temperature and humidity information. A modest alternative for farmers in impoverished nations, the suggested control system is helpful for a low cost automated solar irrigation pumping system. The majority of the time, farmers estimate the soil moisture content while on the field. The suggested technique is more practical because users don't need to visit the field every time they decide whether to use the motor or not. This technique guarantees the least amount of water waste while saving time. Unlike the majority of resistive sensors on the market, which are frequently corroded, the HiGrow sensor module incorporates a capacitive soil moisture sensor. This module is incredibly user-friendly and simple to set up. Only one type of designed soil (potting mix) was taken into account when determining the moisture content limit through output voltage. The soil also included minerals. Therefore, not all types of soil or crops would be allowed to exceed this determined soil moisture limit. The maximum amount of soil moisture varies depending on the crop and the soil. For instance, in order to successfully grow Boro, a type of high-yielding rice that is typically irrigated in Bangladesh, the soil must be completely soaked before the harvesting season. Testing the soil moisture limits before implementing this device is strongly advised. Temperature and humidity boundaries differ from one location to another.

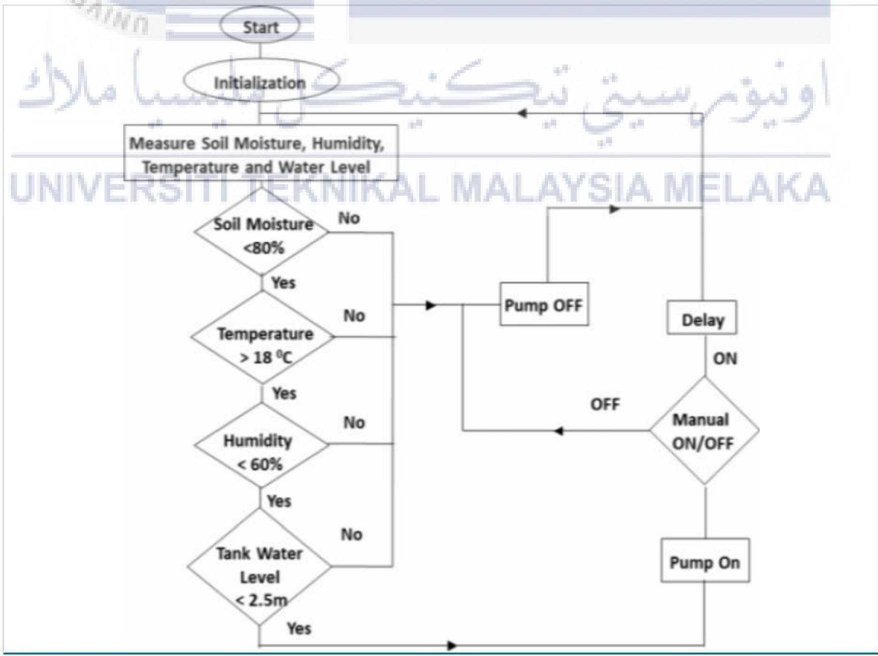


Figure 2.9: Flowchart of Solar Water Pumping System Control Using a Low Cost ESP32 Microcontroller [12]

2.4.10 Automatic Clothing Drying Using Rain Sensor and LDR Sensors On Arduino UNO [13]

Most of the people experience anxiousness while drying their garments during the rainy season. When drying clothing outside the house when the house is not there, this concern will increase. People were reluctant to dry their clothes outside after that happened because they were afraid the clothesline would become wet in the rain. Most individuals dry their clothes on the terrace of their homes during the rainy season since they want to prevent exposing their clothing to the elements while the owner is out doing outside activities. The contributor got the idea to create an automatic clothesline towing system from the description of the issue given above. Applying an Arduino microcontroller, the device a rain sensor and a light dependent resistor sensor are connected to the Uno. When the LDR sensor is not receiving light, the tool interprets this as rain, causing the clothesline to be drawn to a location that is protected from rain. This tool works by detecting the surrounding weather with the rain sensor and LDR sensor. The tool will shift the clothesline to a location where it is exposed to the sun when the sensor detects sunshine since it will interpret this as a sign that it is hot outside. While raindrops are being detected by the rain sensor. It is hoped that the development of an automatic clothesline puller will help people feel less anxious when hanging out their laundry during the rainy season.

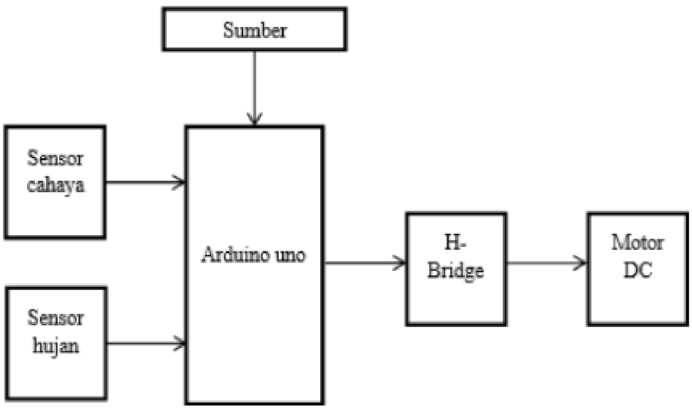


Figure 2.10: Block Diagram [13]

2.4.11 Design And Development of SMART Automated Clothesline [14]

This project aims to create an automated clothesline that can retract and extend each clothesline in response to changes in the weather. The objectives of the project are to create an automatic clothesline that is climate-friendly, use an Arduino microcontroller to operate it, and test a prototype in a real-world setting. Hardware elements included in the prototype model include an Arduino UNO, a rain sensor module, an opto-isolator relay module, an actuator, batteries and a Wi-Fi shield. Programming and mobile app integration are done using BLYNK and the Arduino IDE tools. The rain sensor detects when it starts to rain and lowers the clothesline to prevent the clothes from getting wet. The clothesline is reopened when the weather improves. The project intends to reduce the difficulties people encounter when trying to dry their clothes outside due to bad weather. It focuses on people who live in apartment buildings and those who have difficulty with everyday responsibilities outside of the home. The automatic clothesline makes controlling the drying of clothing flexible, convenient and time saving, allowing people to concentrate on other duties.The Waterfall Model, which includes consecutive phases including requirement analysis, system design, implementation, testing, deployment, maintenance, and assessment, is the approach used in this project. The system needs an Arduino UNO, a water sensor, a servo motor and jumper wires in order to function. The Smart Automated Clothesline project's overall goal is to provide individuals managing their laundry needs with a convenient and effective option for drying garments outside inclement weather.

2.4.12 An IoT Based Real-Time Detection and Notification System for An Economical Solar MPPT Model [15]

The project analyses the application of solar energy as a means of addressing the rising demand for electricity caused by industrialisation and population increase. It draws attention to the difficulties in producing solar power, such as higher initial installation costs and less efficiency. Researchers have concentrated on increasing effectiveness, lowering expenses, and creating maximum power point tracking (MPPT) algorithms to address these issues. By continuously monitoring and running the photovoltaic (PV) array at its maximum power point, MPPT algorithms are utilised to maximise the efficiency of solar power systems. The MPPT algorithms perturb and observe (P&O) and incremental conductance (IC) are two that are frequently employed. Proteus, a platform for automating research and scientific execution, is also mentioned in this project as a tool for modelling and simulating MPPT algorithm performance. A virtual prototype environment similar to hardware prototypes is offered by Proteus. The absence of PV panel models in the Proteus library is a drawback, though it can be overcome by creating a single diode PV panel model in Proteus. The paper gives a general overview of the solar power system description, modelling, MPPT algorithms, sensor descriptions for IoT monitoring, and the integration of Arduino and Proteus for effective operation and performance evaluation.

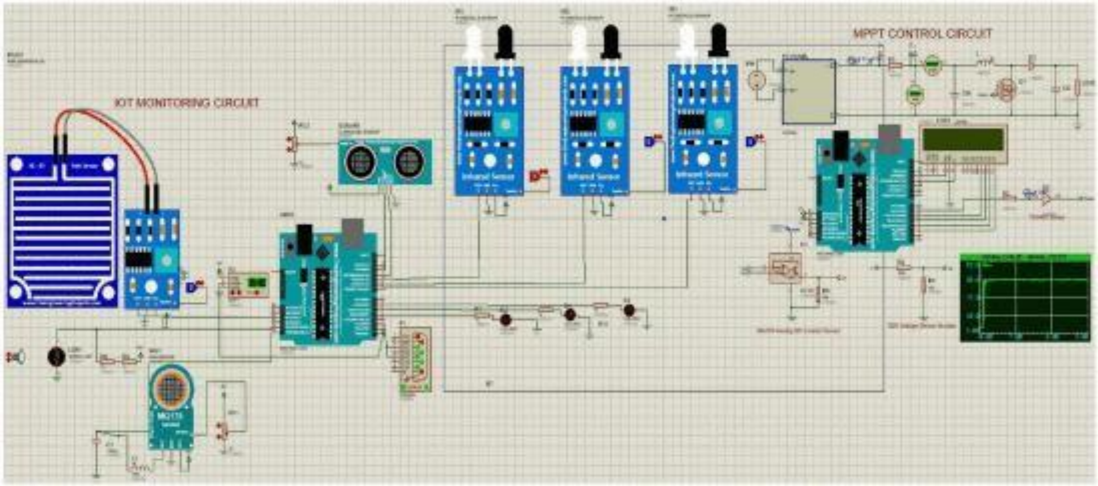


Figure 2.11: The Proposed IoT Based SPV System Designed in Proteus [15]

2.5 Comparison of Previous Related Works

The following table shows a comparison between the previous related works that have been discussed in 2.4 Previous Chapters. This table explains in more detail in terms of source/author, tools/methodology, advantage, and disadvantage in each previous chapter.

Table 2.1: Comparison of previous related works

Source/Author	Tools/Methodology	Advantage	Disadvantage
[4] Ardiansyah, Ahmad Yusuf Sarno, Riyanarto Giandi, Oxsy	• DHT22 • Arduino Uno • Rain Sensor • Fuzzy Logic Method • Fuzzy Control Interfacing Visual Studio 2006	Increasing the accuracy of rain detection by combining temperature, humidity, and rain sensors with a fuzzy logic-based system. Because the Arduino microcontroller is so user-friendly, even non-experts in electronics can easily create the system. With precise rainfall data for several uses, the proposed technology has applications in agriculture, water resource management, and weather forecasting.	The limitations of the rain detection system are mostly controlled by the specs of sensors such as the DHT22, with some restrictions pertaining to temperature range and rain sensitivity under circumstances. Furthermore, fuzzy logic may not be as accurate as other mathematical models or machine learning techniques, even though it is useful for handling imprecise input. This could restrict the system's capacity for understanding complexities and dynamic rainfall patterns.

[5] Gifari, Mohammad Haekal Fahmi, Irfan Thohir, Ajid Syafei, Abdullah Mardiati, Rina Hamidi, Eki Ahmad Zaki	• Light Sensor • Rain Sensor • Humidity Sensors • Smartphone • Actuator (Fan and Motor) • Stopwatch • Pipette	In order to fulfil a variety of uses, including hydroelectric power projects, weather forecasting, agriculture, and water resource management, the suggested rain detection system combines temperature, humidity, and rain sensors. By delivering a user-friendly interface and enabling the smooth integration of hardware and software, the Arduino microcontroller improves accessibility and makes it usable by everyone with a basic understanding of electronics.	The accuracy of the rain detection system is largely dependent on the quality and calibration of key sensors, including the temperature, humidity, and rain sensors. Any errors in these sensors could have a negative impact on the system's overall dependability and performance. The memory, input/output, and processing power limitations of the Arduino microcontroller may further limit the system's functionality and reach, depending on the intricacy of the rain detection system and the applications for which it is designed.
[6] Koushal, Ankit Gupta, Rahul Jan, Farman Kamaldeep, Kamaldeep Kumar, Vikram	• Central Microcontroller Panel • ESP32 Microcontroller • Firebase • Computer or Mobile Devices • Relays	By enabling customers to remotely monitor and operate their appliances and home systems, home automation provides convenience. By doing away with the need to manually manage each item, centralised control maximises efficiency and energy use. In order to create a unique and comfortable living area, customisable scenarios, such a greeting routine, automate operations like lighting adjustments, thermostat settings, and music activation.	A complete home automation system's implementation might be expensive because of the costs involved in purchasing the electronic devices, sensors, controllers, and hardware that are required. Furthermore, some systems that depend on internet connectivity for remote access could experience problems in places with spotty or unstable network access, which could be a disadvantage for users who depend on reliable remote control capability.

[7] Mohd, Nik Hashim, Zarifie Afiq, Muhammad Adnan, Asyraf Sulistiyo, Mahmud Dwi	• Rain Sensor • Microcontroller • Motor • Buzzer • Power Supply	By automatically retracting the clothesline when rain is detected, the automatic clothesline puller provides convenience by removing the need for work in laundry control. Users can concentrate on other tasks while this time-saving technology ensures that laundry is dry and ready to use by keeping items from becoming wet and possibly damaged.	The rain detection device can only detect rain and pull the clothesline back in reaction to it; it cannot detect other weather conditions that could damage laundry, such as wind, dust, or too much sunlight. The dependability and effectiveness of the system could be jeopardised by inaccurate readings from the rain sensor, which could cause the clothesline to activate prematurely or not to retract during real rain.
[8] Nur Aisyah Abdul Hei, Siti Nadzirah Nazri, Effa Faqihah Mohamed Rafik, Nurin Berahim, Mazniha Tun Hussein Onn Malaysia, Universiti KM Panchor, Jalan	• Arduino IDE • Humidity Sensor • Alarm System • Motor	The requirement for manual weather and clothing control is eliminated by automated sensors. In order to save customers time and effort, the system proactively detects rain and takes the appropriate measures, such as moving clothing to a dry place. This laundry management optimisation increases the efficacy of daily life by relieving users of weather-related chores and freeing them up to concentrate on other activities.	The efficiency of the rain detection system depends on the humidity and rain sensors providing accurate readings; in the event that these sensors give inaccurate information, there may be false alarms or inefficiencies in the system. The automated washing management system's total cost-effectiveness and affordability are important factors to take into account. These include the costs associated with sensors, microcontrollers, motors, and continuous power sources. It may also be necessary to have backup plans in place to ensure that the system continues to function even in the event of power outages or electrical malfunctions.

[9] Ton Mohamad, Mohamad Ihsan	• Water Level Sensor Circuit • PIC Microcontroller • Pulley and String • 12V DC Motor	The automated laundry management system saves users time as well as effort by managing their laundry on its own without involving them directly. This ensures that clothing is protected from rain, saves users time, and may not require drying or rewashing. It also does away with the necessity for continuous weather monitoring. In order to accommodate different tastes and weather circumstances, the system provides customers with the option to dry their clothes in an open space or in a secure location.	With potential problems emerging from false negatives or false positives in the detection process, accurate rain detection is essential for eliminating unwanted clothes movement or rain exposure in the automated system. The system must first be configured by buying and installing a number of hardware components, which is more expensive and requires technical know-how. The automated rain detecting system requires regular maintenance, troubleshooting, and potential software updates to guarantee its long-term efficacy.
[10] Siti Dalilah Hashim, Muhamad, and Nor Masharah Husain	• KG004 Rain Sensor Module • FC-37 Rain Sensor • GSM SIM800C Module • Arduino Uno R3 • Arduino IDE • Buzzer • ESP8266 • Water Level Detector • ThingSpeak	Real-time meteorological information is provided by sensors and IoT technologies, which continuously monitor water levels and rainfall. With the help of reasonably priced parts like the ESP8266 and KG004 rain sensor module, the system functions as an efficient and economical flood or natural disaster warning system. Users may easily evaluate and monitor the collected data with the accompanying smartphone application, allowing for prompt preventive action.	The accuracy of the data obtained may be impacted by the rain sensor's inability to distinguish between light drizzle and rainfall due to its low detection threshold of 0.5mm/h. Furthermore, the project's limited flexibility and scalability to other locations are caused by its geographical focus on Peninsular Malaysia's east coast, necessitating adjustments and extra sensors for expansion. In order to avoid system disruptions or inaccurate data, it is essential that the sensors, cloud platform, microcontrollers, and mobile application all have secure connections.

[11] Otchere, Peter	• Ran Sensor • Wiper Motor • Control Board • AC supply • Transformer • Rectifier and filter • Infrared Light Source.	By detecting water, the automated rain sensor system turns on the windscreen wiper, lowering the possibility of visibility-related incidents and improving road safety. With no human control needed, the automated wiper system improves comfort and frees up the driver to concentrate more on the road. Its energy-efficient design also prolongs the life of the car's battery and reduces the pressure on the electrical system.	For the automatic rain sensor system to function effectively, accurate water detection on the windscreen is essential. This prevents inconsistent or false-positive wiper activation and ensures that wiper activation is timed precisely. The system's usefulness, however, would be constrained because it only targets water and might not react as well to other substances that reduce vision, including dirt, snow, or ice, requiring manual interventions in those circumstances. Furthermore, the overall health of the vehicle's electrical system is necessary for the automated rain sensor system to operate properly; any problems here could also affect how well the wiper system works.
[12] Bipasha Biswas, Shatadru Tariq Iqbal, M.	• ESP32 • Programmable Sensor Module • Solar PV • Solar PV Array • Water Level Sensor • Motor Pump Set • Inverter • HOMER (Hybrid Optimisation and Multiple Energy Resources)	The affordable automated solar water pumping system is intended for use in underdeveloped areas, offering farmers in those places useful and accessible irrigation equipment. Thanks to the ESP32 microcontroller and web server integration, the system's temperature, humidity, and soil moisture sensors optimise irrigation techniques, minimise water waste, and enable remote management. This gives users the convenience and flexibility to monitor and control the system from any location using mobile devices.	Since the set moisture constraints might not apply to all soil types and crops, farmers should evaluate the applicability of the soil moisture system for their particular needs. Even though solar-powered irrigation systems are economical and eco-friendly, their effectiveness can vary depending on local climate factors and sunshine availability. This could lead to lower system reliability in regions with little sunlight or a lot of cloud cover.

[13] Athaya Atsiq Andryan Gunawan, Amin Alqudri Dwi Nugraha	• Rain Sensor • Light Dependent Resistor (LDR) Sensor • Arduino Microcontroller • Actuator or Motor • Power Supply • Wiring and Connectors • Pulleys • Cables • Gears	In order to save users time and effort, the automatic clothesline pulling system has a rain sensor built in to make sure the clothing on the line stay dry. The system also makes use of a light-dependent resistor (LDR) sensor to sense sunshine and alter the position of the clothesline for the best possible sun exposure, giving users confidence in the efficient drying of their clothing even when they are away from home.	There is a lack of information in the project on the power source selection for the Arduino microcontroller and other components could cause problems with accessibility and maintenance. The automatic clothesline pulling system can be difficult to use for people who are not familiar with electronics or programming because it demands technical competence in Arduino programming. The rain and LDR sensors' accuracy and dependability are critical to the system's overall performance; they can be inaccurate in certain weather scenarios, which could result in system problems.
[14] Nasrulddin, Mohd Latif, Abd Abd Aziz, Ashykin Ramdan, Mohamad Rohieszan Othman, Hafiza	• Rain Sensor Module • Arduino UNO • Opto-isolator Relay Module • Actuator • Batteries • Wi-Fi shield • BLYNK • Arduino IDE • Water Sensor	When rain begins, the automated clothesline enters operation to keep clothes dry and promote resource and energy efficiency by preventing unnecessary drying cycles and lowering dependency on energy-intensive alternatives like clothes dryers. Because the system operates autonomously, customers can focus on other tasks knowing that their clothing will be properly covered and dried when the weather improves. This eliminates the need for manual weather monitoring.	Specific hardware is needed for the automated clothesline, such as an Arduino UNO, an actuator, an opto-isolator relay module, rain sensor module, batteries, and a Wi-Fi shield. However, some customers—especially those with little technical experience—may find it difficult to afford and set up these components due to their technical complexity and accompanying expenses. For the system to be successfully implemented and used, adequate help and documentation are essential, especially for those who are not as experienced with hardware integration, Arduino programming, or mobile app integration.

[15] Katyal, Aanchal Pathak, Diwaker Gaur, Prerna	• Maximum Power Point Tracking (MPTT) • Proteus • Arduino Sensors • Solar Panels • Single Diode PV Panel Model	The project's main goal is to use solar energy to meet the expanding need for power brought on by population growth and industrial development. The research suggests employing maximum power point tracking (MPPT) techniques like perturb and observe (P&O) and incremental conductance (IC) to increase overall system efficiency in recognition of the constraints of solar energy efficiency. Researchers can save time and resources by evaluating and optimising the MPPT algorithm's performance in a virtual environment prior to physical implementation by using Proteus as a modelling and simulation platform.	The widespread adoption of solar power systems may be hampered by high installation costs, particularly in places with low financial resources. Current solar technologies are less efficient than traditional energy sources, even with efforts to address economic problems. However, there is room for improvement, especially when compared to conventional power generating techniques. Furthermore, the lack of pre-built PV panel models in the Proteus library could make research more difficult since it would force researchers to create their own models, which could lead to inaccuracies.

2.6 Summary

The literature review "Development of an IoT-based Smart Clothesline System Using Rain Detection Sensors" examines developments in the field of smart clothing and Internet of Things (IoT) technology to improve clotheslines' functioning. Based on an extensive literature review and thorough comparisons conducted, this project will utilize a variety of sophisticated hardware and software tools. I will give a thorough explanation and description of the carefully chosen tools that will be applied to complete this project in the following chapter.

CHAPTER 3

METHODOLOGY

3.1 Introduction

The methodology section titled "Development of an IoT-Based Smart Clothesline System Using Rain Detection Sensors" describes the systematic method used to design, implement, and test the suggested smart clothesline system. This section's major goal is to give users a clear understanding of the method used to design systems step by step. The hardware employed, such as rain detecting sensors, microcontrollers, and alerting systems, are thoroughly explained. Moreover, the method applied to construct the software includes the programming language and framework.

3.2 System Design

3.2.1 Block Diagram

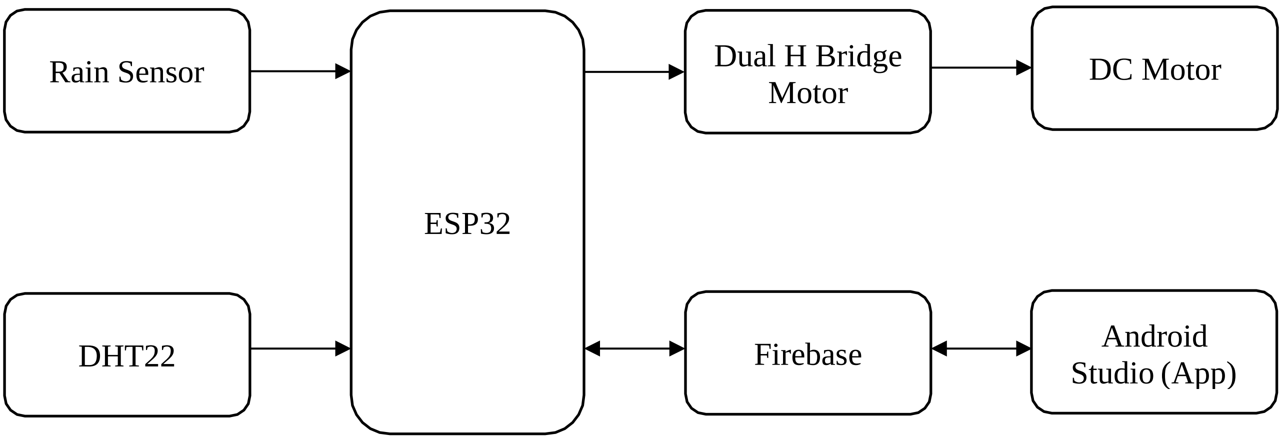


Figure 3.1: Block diagram system

Block diagram above shows the design concept of the “Development of IoT based Smart Clothesline System using Rain Detection Sensor”. The rain sensor is used to detect when it's raining. It can sense the presence of raindrops or moisture and sends a signal when rain is detected. In addition, a humidity sensor measures how much moisture or humidity is in the air. It provides data on the level of moisture in the environment, like how humid or dry it is. Therefore, a humidity sensor primarily serves to monitor and gather data regarding humidity levels, whereas a rain sensor is designed mainly for detecting rain.

3.2.2 Use Case Diagram

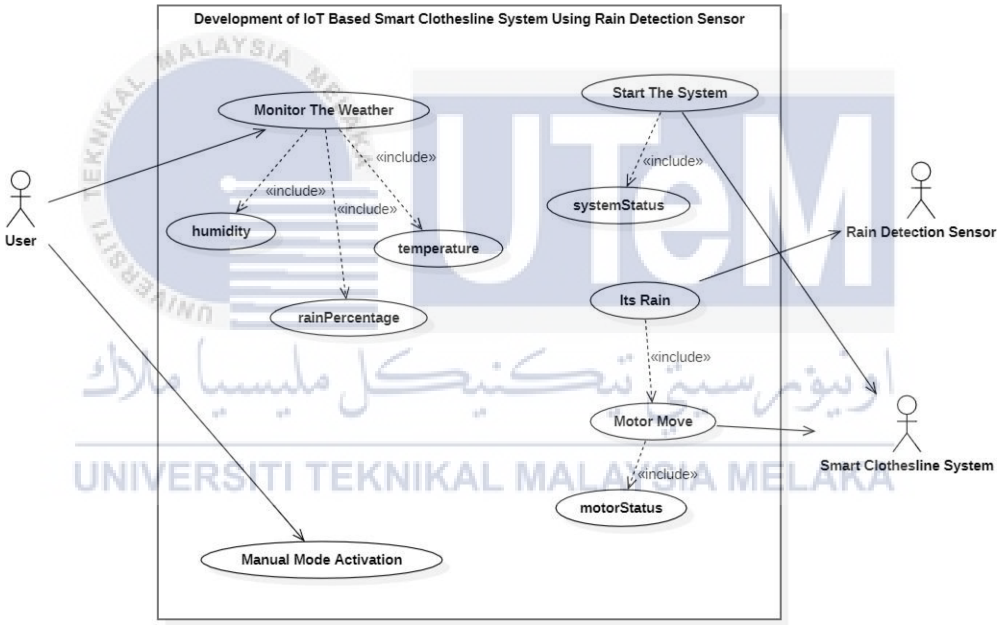


Figure 3.2: Use case diagram of Smart Clothesline System

User interacts with the system through manual controls or monitors whether conditions through the Android application in smartphone. Rain detection sensor interacts with rain detection to monitor the raindrop. Smart clothesline system checks if the rain percentage is more than or equal to 30%, the motor status is on and moves for 3 seconds and automatically stops the motor.

3.2.3 System Flowchart

The following figure shows the system of ESP32 controller.

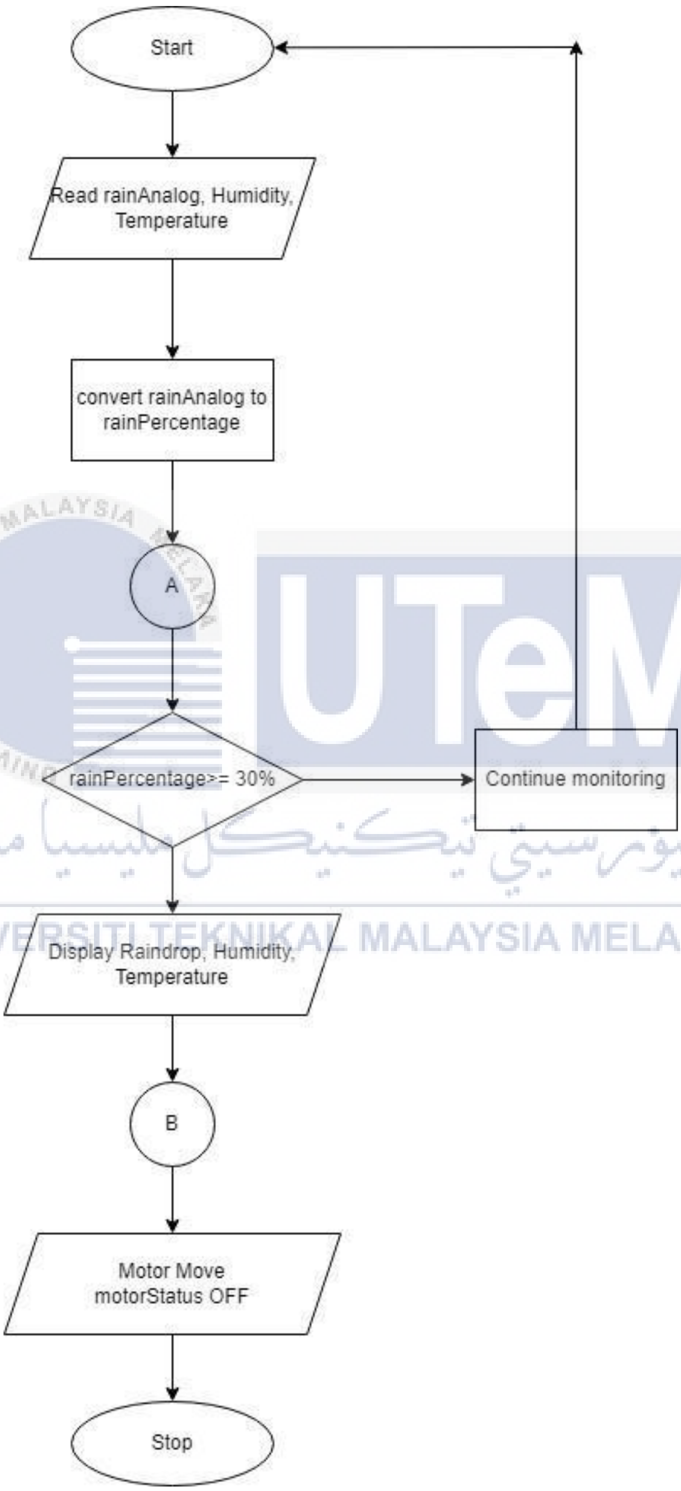


Figure 3.3: Flowchart of Smart Clothesline System

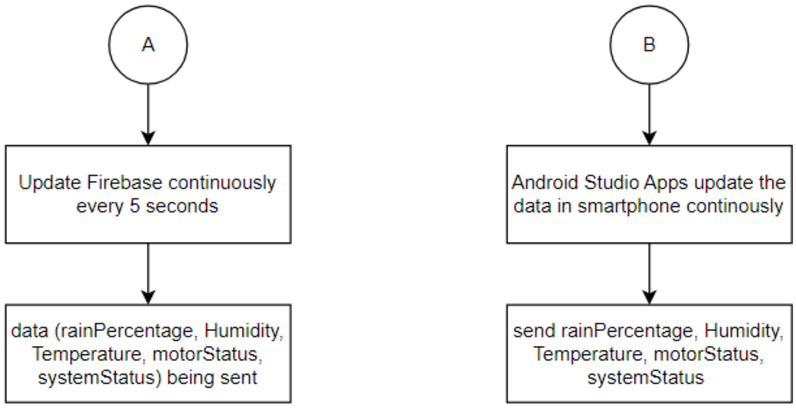


Figure 3.4: Operational Flow of The Smart Clothesline System

This system begins when the Rain sensor, Humidity and Temperature sensor reads the data from the presence of water and the existing temperature in the environment. The data of raindrop was read in analog, so the data will convert to rainPercentage. The data will update to Firebase continously for every 5 seconds. Data received from the sensor will be grouped and if the rainPercentage is greater and equal to 30%, it will be triggered to activate the sensor and indicate the presence of rain. These sensors are designed to be highly sensitive and can detect even small amounts of moisture. Based on the previous literature review, when the rain sensor reaches a 30% activation level, the output response indicates drizzle [2]. To avoid getting the clothes wet, it is necessary to promptly remove them. It also update the data in Android Studio (app) in smartphone. When the system detects the rain then the DC motor to activate and initiate movement. Once the motor moves, the motor will be automatically stop for moving. Rain percentage, temperature and humidity data will continuing displayed in real-time on the mobile phone. However, if the water drop is below the 30%, the DC motor will remain stationary and continue monitoring for any water drop occurrence.

3.2.4 Android Studio Design

The following figure shows a sketch of an Android Studio application. There are two interfaces that have been designed for the Smart Clothesline System. The first interface is the front page after entering the application. there is a textview that says "Welcome To Smart Clothesline Application" with a background image behind. There are also two toggle buttons for "Keep Monitor" and "Manual Button". The second interface shows a textview to the value of rain in percent, temperature in degrees Celsius and humidity in percent. This interface also comes with a background image on the back.

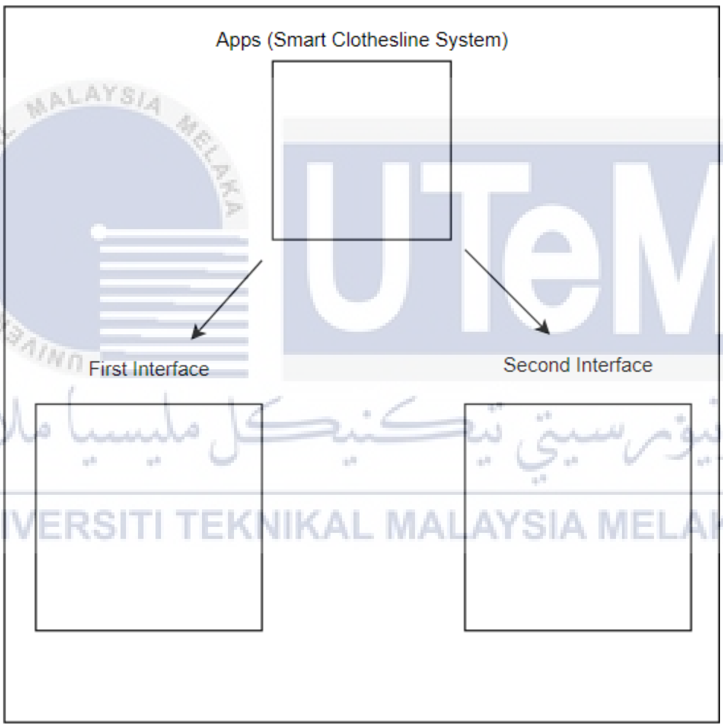


Figure 3.5: System application sketch

3.3 Gantt Chart

Table 3.1: Gantt Chart table BDP1

Activity	Week													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
BDP1														
Determination of project title														
Registration of project title														
Online meeting with JK PSM														
Meeting with supervisor														
Identify the scopes and objectives														
Do a research about hardware and software related to rain detection														
Design the concept of system for remove clothes														
Development of hardware and software														
Testing the design														
Diagnose and record the design development														
Make an analysis														
Writing project report														
Submission project report														
Presentation														

Table 3.2: Gantt Chart table BDP2

Activity	Week													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
BDP2														
Design a gantt chart														
Meeting with JK PSM														
Meeting with supervisor														
Assembly DHT22 connection														
Connect and verify DHT22 connection														
Testing on DHT22														
Assembly rain sensor connection														
Connect and verify rain sensor connection														
Testing on rain sensor														
Asembly L298N with DC motor connection														
Connect and verify L298N with DC motor connection														
Testing on L298N with DC motor														
Assembly a full connnection														
Connect and verify full connection														
Testing on full connection														
Develop a mobile application (GUI)														
Connect hardware with Firebase														
Connect software with Firebase														
Develop a prototype														
Writing project report														
Design poster for presentation														
Presentation														
Submission project report														

3.4 Hardware Tools

3.4.1 NodeMCU ESP32

The ESP32 generation of system on a chip microcontrollers features integrated dual-mode Bluetooth and Wi-Fi and is inexpensive and low power. The Tensilica Xtensa LX6 dual-core or single-core microprocessor, Tensilica Xtensa LX7 dual-core, or a single-core RISC-V microprocessor are used in the ESP32 series, which also has integrated antenna switches, RF baluns, power amplifiers, low-noise receive amplifiers, filters, and power-management modules [14] ESP32 can function as a full independent unit or as a slave device to an embedded MCU, which lessens the burden on the primary application CPU caused by communication stack overhead [15]. The GPIO (General Purpose Input/Output) on NodeMCU is accessible, and the API documentation includes a pin mapping table.

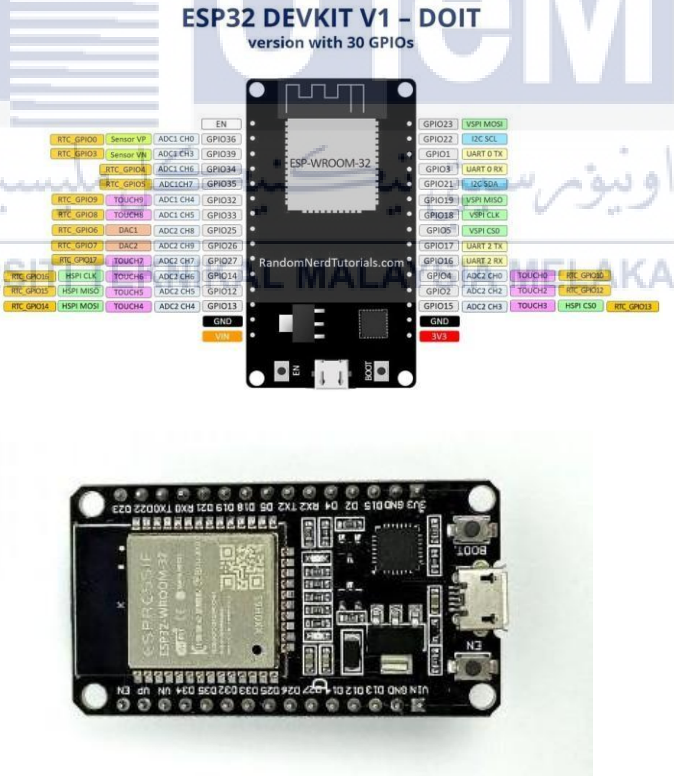


Figure 3.6: Pin Out of NodeMCU ESP32 [18]

3.5 Rain Sensor

One type of switching device used to detect rainfall is a rain sensor. The functioning mechanism of this sensor is similar to that of a switch; if it rains, the switch is typically closed. The resistance principle underlies the operation of the nickel coated lines on the rain sensor board. This sensor module allows for the measurement of moisture via analogue output pins and produces a digital output when the moisture threshold is exceeded [16].

Figure 3.7: Rain Sensor [19]

3.6 DHT22 [2]

With a single wire digital interface, the DHT22 is an affordable digital temperature and humidity sensor. It measures the surrounding temperature using a thermistor and a capacitive humidity sensor before emitting a digital signal on the data pin (no analogue input pins are required). The sensor is calibrated and requires no additional parts, allowing you to accurately measure temperature and relative humidity.

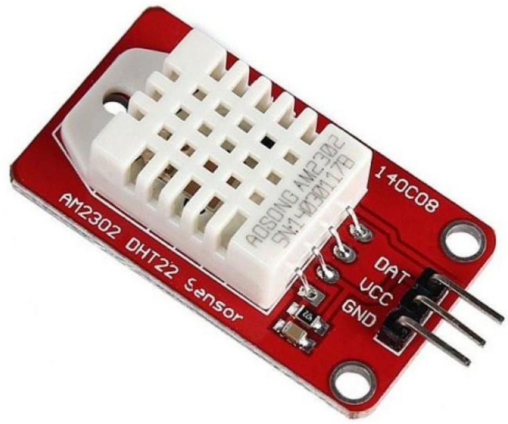


Figure 3.8: DHT22 [2]

3.7 5V DC Motor

Any group of rotating electric motors that use direct current (DC) electricity to create mechanical energy is referred to as a DC motor. The majority of types rely on the magnetic field's forces. For a portion of the motor's current to occasionally shift direction, almost all types of DC motors contain an internal mechanism that is either electromechanical or electronic. Because they could be supplied by existing direct-current lighting power distribution networks, DC motor was the first type of motor that was widely employed. A DC motor's speed can be modified across a large range by adjusting the supply voltage or the amount of current flowing through its field windings [17].

Figure 3.9: 5V DC Motor [21]

3.8 Dual H Bridge Motor Driver L298N

L298N Dual H-Bridge Motor Controller regulates motor rotation and speed. In addition, it can be used with other goods including LED arrays, relays, and solenoids. It has a strong heat sink and a tiny motor driver that is rather powerful. able to supply a maximum of 2A to motors between 5V and 35V [18].

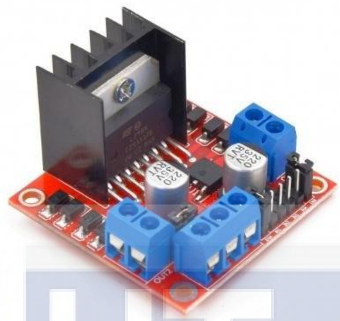


Figure 3.10: Dual H Bridge Motor Driver L298N [22]

3.9 Software Tools

3.10 Arduino IDE [19]

A full-featured integrated development environment that enables programmers to create, assemble, and run Android apps on their Android-powered devices. Java-only console applications are also supported by AIDE, along with the development of apps using C++ and the Android NDK and Java/XML with the Android SDK. AIDE and Eclipse projects work together flawlessly.

3.11 Android Studio

The official Integrated Development Environment (IDE) for creating Android apps is called Android Studio. Based on IntelliJ IDEA's robust code editor and developer tools, Android Studio provides even more features to increase your efficiency when developing Android apps, such as a configurable build system based on Gradle. A quick emulator with many features allows all Android devices to be developed in a single environment, including live editing to instantly update composables on actual devices and in emulators. You may import sample code and construct typical app features with the aid of code templates and GitHub integration. Extensive testing frameworks and tools, using linters to detect issues with performance, usability, version compatibility, and other issues NDK support, C++ support and support for Google Cloud Platform are pre-built [20].

3.12 **Firebase** [21]

Firebase is a Google mobile application development platform with strong tools for creating, managing, and upgrading apps. A backend platform for creating online and mobile applications is called Firebase. Fundamentally, Firebase is a set of tools that developers can use to build applications and scale them in response to demand. Firebase seeks to address three key issues for developers: how to design an app, deliver it quickly, confidently monitor it and engage users. By using this platform, developers can access services that they otherwise would have to create themselves and instead concentrate on providing engaging application experiences. The Google Firebase platform has several noteworthy capabilities, including databases, authentication, analytics, push notifications, file storage and much more. Developers may easily do on-demand scaling because the services are hosted in the cloud. Currently, developers all over the world rely on Firebase, one of the top platforms for developing mobile applications.

3.13 Summary

Many important steps are described in this chapter. First, the hardware parts are chosen and configured. These include the rain detection sensor, ESP32 microcontroller and DHT22. For real-time rainfall monitoring, the rain detection sensor is calibrated and attached to the microcontroller. In order to enable data collection from the rain sensor and decision-making based on the detected rainfall, software programming for the ESP32 is then built. The notifications will immediately send to their mobile device through the system logic. In order to ensure that the smart clothesline system operates effectively and efficiently, the complete system is tested and determined for its precision, reactivity and overall performance in sensing rain and sending timely messages.

CHAPTER 4

RESULTS AND DISCUSSIONS

4.1 Introduction

This chapter presents the results and analysis of "Development of IoT Based Smart Clothesline System Using Rain Detection Sensor". The system has been implemented in smart clothesline system. The aim of this project is to strengthen the drying of clothes at the clothesline by applying IoT technology with a sensor that detects rain. By detecting rainfall in a real-time database, the technology aims at helping users with timely notifications, enabling them to let the DC motor pull it or get clothes manually and stay dry. This results show the rain detection sensor has been successfully integrated with the IoT framework, demonstrating the system's potential to improve convenience and effectiveness in drying clothes.

4.2 Results and Analysis

The table shows the value of rain percentage based on rain drop value. The value of rain drop starting from 0mL to 2mL and the value of rain percentage ranges starting from 0% to 72%. As we can see, the rain drop value increases, the rain percentages also increase.

Table 4.1: Rain percentage based on rain drop value

Rain Drop (mL)	Rain (%)	Rain (Analog Value)
0	0	4095
0.2	34	2702
0.4	50	2047
0.6	51	2007
0.8	54	1884
1	57	1761
1.2	63	1561
1.4	66	1393
1.6	68	1311
1.8	69	1270
2	72	1147

The graph shows the rain percentage value based on rain drop (mL) over rain percentage. The rain percentage increase if rain drop increase. As we can see, the graph increases lineary.

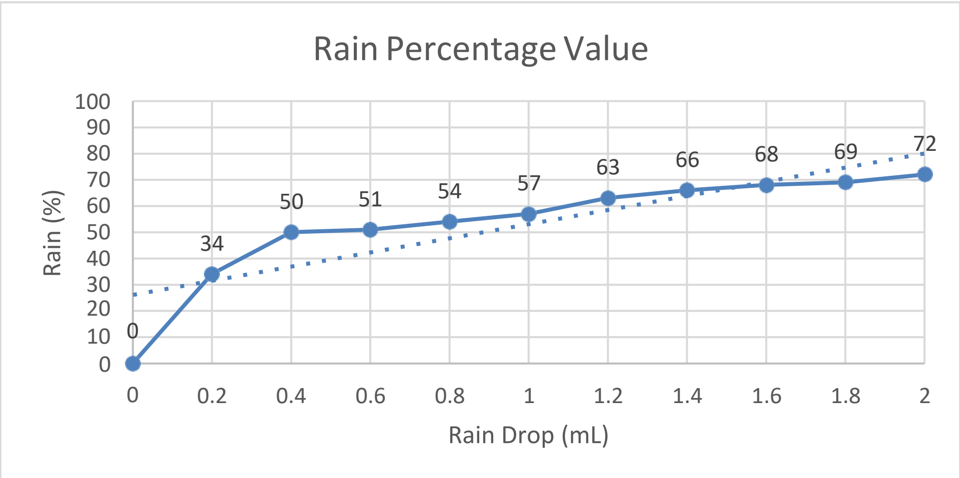


Figure 4.1: Graph of rain percentage value

The graph shows the rain (analog value) based on rain drop over analog value of rain. 4095 is a lower boundary since it reads a completely dry condition (0%) and 0 is an upper boundary, reads as a completely wet condition (100%).

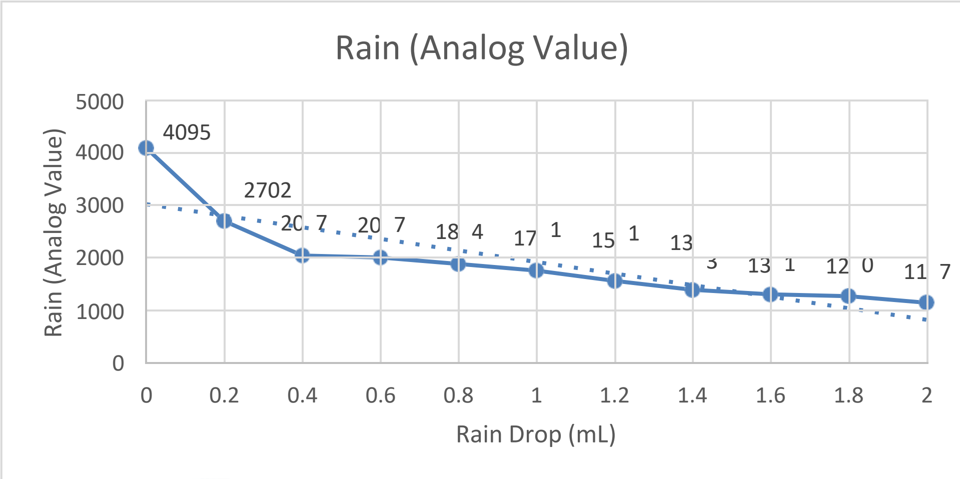


Figure 4.2: Graph of rain analog value

The figure shows data analysis in Arduino IDE. The delay for the system is just 5 seconds which u can refer the timestamp above.

```
01:38:42.326 -> Humidity: 89.10%, Temperature: 29.20°C
01:38:42.369 -> Rain Sensor Value: 4095
01:38:42.391 -> Rain: 0%
01:38:42.433 -> Motor Stop2
01:38:47.341 -> Humidity: 89.10%, Temperature: 29.20°C
01:38:47.370 -> Rain Sensor Value: 2495
01:38:47.396 -> Rain: 39%
01:38:47.435 -> Motor move
01:38:50.352 -> Motor stop
01:38:55.351 -> Humidity: 89.10%, Temperature: 29.20°C
01:38:55.393 -> Rain Sensor Value: 1950
01:38:55.414 -> Rain: 52%
01:38:55.414 -> Motor Stop2
01:39:00.362 -> Humidity: 89.00%, Temperature: 29.20°C
01:39:00.403 -> Rain Sensor Value: 2109
01:39:00.428 -> Rain: 48%
01:39:00.428 -> Motor Stop2
```

Figure 4.3: Data analysis in Arduino IDE

4.3 Hardware Setup

The following figures show the prototype that was designed in Chapter 3. The first figure is the prototype from above where the rain sensor module is placed on the roof of the house to detect raindrops. The second figure shows the DHT22 placed on the side of the house to read the humidity and temperature in the surrounding area of the house. A 5V DC motor is placed inside the covered area to pull the suspension from the uncovered area into the covered area.

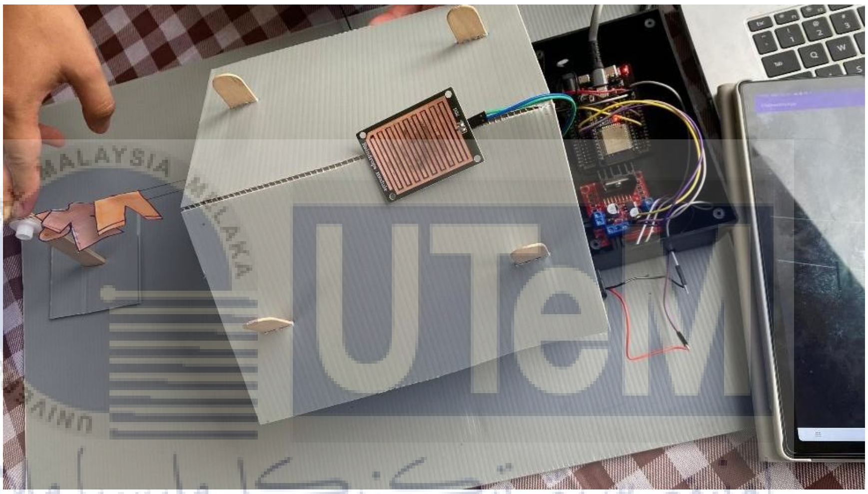


Figure 4.4: View of prototype from above

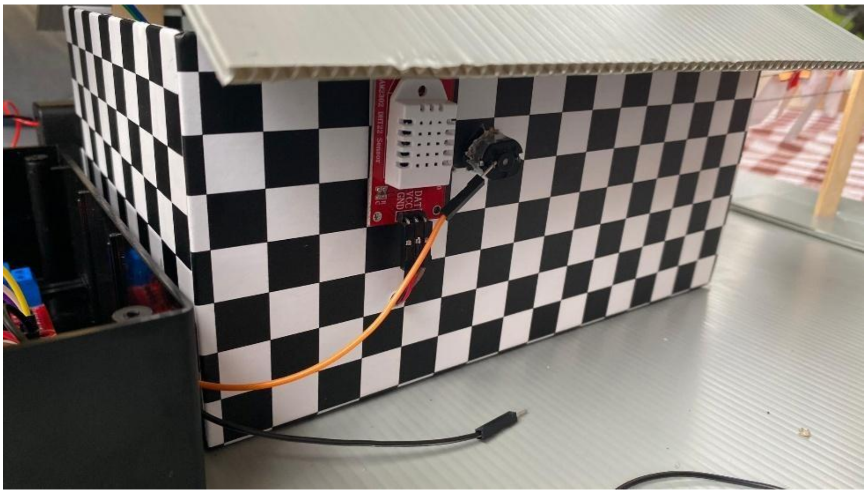


Figure 4.5: View of prototype from side

4.4 Software Setup

Code listing for manualButton (Figure 4.6), when the button was clicked, it triggers the 'onCheckChanged' method. When systemStatus is set to 0, the motor will stop moving while systemStatus is set to 1, the motor will start functioning until the rain sensor triggered. The ESP32 control the motor based on Firebase database. Motor will triggered when it notices a change in the apps. Code listing, systemStatus (Figure 4.7) for manual mode by user, connect with Firebase. User need to click the button when they want to turn on manual mode.

```java
manualButton.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {
    @Override
    public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
        int newSystemStatus = isChecked ? 0 : 1; // Assuming 1 is system on, 0 is system off
        databaseReference.child( pathString: "systemStatus").setValue(newSystemStatus);

        if (newSystemStatus == 0) {
            finish();
        }
    }
});
```

Figure 4.6: Code segment (manualButton) MainActivity.java in Android Studio

```java
@Override
public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
    // Retrieve system status from the database
    Integer systemStatus = dataSnapshot.child( path: "systemStatus").getValue(Integer.class);

    if (systemStatus != null) {
        // Update the manual switch state based on the retrieved value
        updateManualSwitchState( systemStatus: systemStatus == 1); // Assuming 1 is system on, 0 is system off
    } else {
        Log.e( tag: "MainActivity", msg: "System status is null");
    }
}
```

Figure 4.7: Code segment (systemStatus) MainActivity.java in Android Studio

Based on the proposed display page of interface at Figure 3.3 has successfully generated at Figure 4.8 and Figure 4.9.

Figure 4.8: First interface (activity_main.xml)

Figure 4.9: Second interface (activity_display.xml)

Code listing, sensors path from Firebase. The value will be retrieved from the database. Then, it updates a value in texview (display_activity.xml).

```java
// Set up ValueEventListener to listen for changes in the database
ValueEventListener valueEventListener = new ValueEventListener() {
    @Override
    public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
        // Retrieve values from the database
        float tempValue = dataSnapshot.child( path: "temperature").getValue(Float.class);
        float humidityValue = dataSnapshot.child( path: "humidity").getValue(Float.class);
        int rainValue = dataSnapshot.child( path: "rainPercentage").getValue(Integer.class);

        // Update TextViews
        updateProgressText(rainProgress, rainValue);
        updateProgressText(tempProgress, tempValue);
        updateProgressText(humidityProgress, humidityValue);
    }
```

Figure 4.10: Code listing Display.java in Android Studio

The GUI, Arduino IDE and Realtime Database when the rain detection sensor did not triggered because no rain occurs.

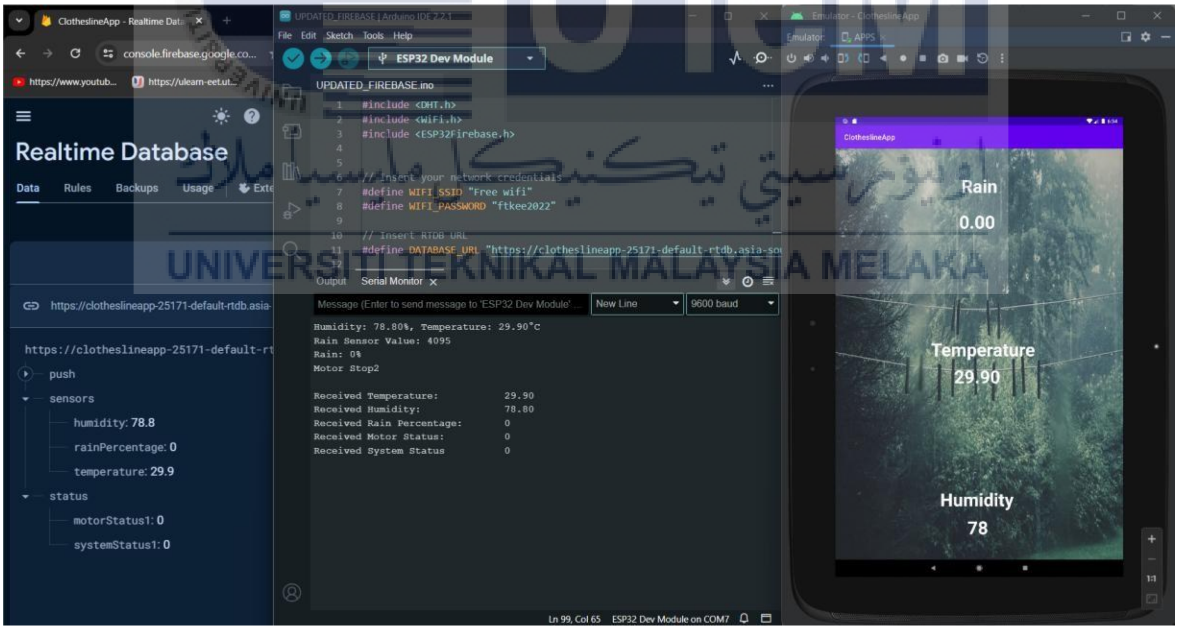


Figure 4.11: The GUI when the Rain Percentage = 0%

The GUI, Arduino IDE and Realtime Database when the rain detection sensor triggered when the rain is present.

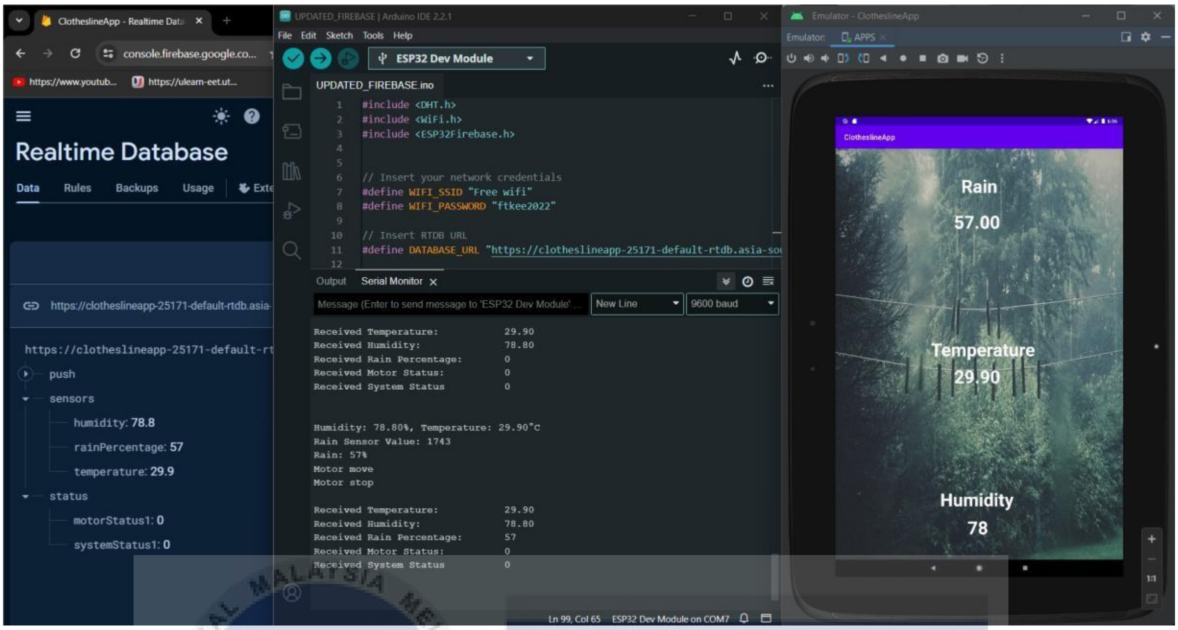


Figure 4.12: The GUI when the Rain Percentage = 57%

The code listing when the condition for rainPercentage >= 30, DC motor will move. Once the motor moved, the motor will stop but still in monitoring humidity and temperature.

```
if ((rainPercentage >= 30) && (motorStatus == 0) && (systemStatus == 1)) {
    // It's raining, turn on the motor for 3 seconds
    digitalWrite(MOTOR_ENA, HIGH);
    digitalWrite(MOTOR_IN1, HIGH);
    digitalWrite(MOTOR_IN2, LOW);
    Serial.println("Motor move");
    delay(3000);

    // Motor stop for 3 seconds
    digitalWrite(MOTOR_ENA, LOW); // Turn off the motor
    digitalWrite(MOTOR_IN1, LOW);
    digitalWrite(MOTOR_IN2, LOW);
    motorStatus = 1;
    Serial.println("Motor stop\n");

  } else {
    // Motor did not move
    digitalWrite(MOTOR_ENA, LOW);
    Serial.println("Motor Stop2\n");
  }
  updateFirebase(temperature, humidity, rainPercentage, motorStatus, systemStatus);
  delay(5000);  // Delay for 5 seconds
}
```

Figure 4.13: Code listing when the rainPercentage triggered

The code listing for Firebase update for temperature, humidity, rainPercentage, motorStatus and systemStatus.

```
void updateFirebase(float temperature, float humidity, int rainPercentage, int motorStatus, int systemStatus) {

  // Update temperature
  firebase.setFloat("sensors/temperature", temperature);
  float temperature2 = firebase.getFloat("sensors/temperature");
  Serial.print("Received Temperature:\t\t");
  Serial.println(temperature2);

  firebase.setFloat("sensors/humidity", humidity);
  float humidity2 = firebase.getFloat("sensors/humidity");
  Serial.print("Received Humidity: \t\t");
  Serial.println(humidity2);

  firebase.setInt("sensors/rainPercentage", rainPercentage);
  int rainPercentage2 = firebase.getInt("sensors/rainPercentage");
  Serial.print("Received Rain Percentage: \t");
  Serial.println(rainPercentage2);

  //firebase.setFloat("sensors/temperature", temperature);
  int motorStatus1 = firebase.getInt("status/motorStatus1");
  Serial.print("Received Motor Status:\t\t");
  Serial.println(motorStatus1);

  int systemStatus1 = firebase.getInt("status/systemStatus1");
  Serial.print("Received System Status \t\t");
  Serial.println(systemStatus1);
  Serial.println("\n");

}
```

Figure 4.14: Code listing for Firebase update

4.5 Summary

This chapter presented case studies to the "Development of IoT-Based Smart Clothesline System Using Rain Detection Sensor" demonstrate the successful integration of a rain detection sensor with an IoT framework. Through a smartphone application, the system efficiently detects rainfall and notifies the user in real-time. According to results, the system accurately detects rainfall and issues applicable alerts, enabling users to take quick action to prevent their clothing from being wet with the successful development of a prototype clothesline system based on the objectives of this project. These early findings establish the viability and efficiency of the IoT-based smart clothesline system and create the groundwork for further advancement and technology improvement.

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

In conclusion, the first objective of this project, which was to perform literature review of clothesline system using rain detection sensor, was successfully accomplished in Chapter 2 (Literature Review). We have learned a lot about the creation of an IoT-based smart clothesline system by looking at the technology elements and design issues, as well as by analysing relevant research and advances. By offering real-time monitoring of rainfall, the incorporation of a rain detection sensor into the clothesline structure has the potential to revolutionise the conventional technique of drying clothes. The development of the prototype which second objective has been achieved in Chapter 3 (Methodology). This current prototype offers a concrete example of the system's operation and creates and outline for future versions. Furthermore, the last objective which is involves analyzing the accuracy and relibility of the system also has been fulfilled throught the measurement of the rain sensor, DHT22 in Chapter 4 (Results and Discussion). Furthermore, by achieving these three objectives, the project has been accomplished of Development of IoT Based Smart

Clothesline System Using Rain Detection Sensor that efficiently senses rainfall, regulates the drying process and includes a user-friendly mobile application interface.

5.2 Future Works

For future improvements, "Development of IoT Based Smart Clothesline System Using Rain Detection Sensor" results could be enhanced as follows:

i) Include a notification system within the mobile app. This improvement would improve user engagement by easily informing users about changing weather conditions and offer an easy way for proactive monitoring of the clothes drying process.Advanced weather prediction algorithms and APIs can be integrated to produce forecasts that are more precise, enabling the system to respond proactively to incoming rain and take preventive action.

ii) The system can adapt drying times and offer specific suggestions by using machine learning and artificial intelligence algorithms. The system can learn from weather patterns, user preferences, and historical data.

iii) Other environmental sensor such as UV sensor can be integrated in addition to rain detection sensors to provide a comprehensive understanding of the drying conditions and to further enhance the drying process in light of numerous environmental elements.

iv) Users now have greater control and flexibility by monitoring the drying process, making settings changes remotely and receiving in-depth analytics and insights on energy usage and clothing care with an improved mobile application interface and functionality.

REFERENCES

[1] "Laundry Drying on Outdoor Clothes Line," *PhotoDune*. https://utemedu-my.sharepoint.com/personal/b082010004_student_utem_edu_my/Documents/FYP/Literature%20Review (accessed Jun. 22, 2017).

[2] DHT22 - digital temperature and humidity sensor (no date) CityOS Air. Available at: https://cityos-air.readme.io/docs/4-dht22-digital-temperature-humidity-sensor (Accessed: 31 December 2023).

[3] DID, "Laporan Banjir 2019," *Dep. Irrig. Drain. Malaysia*, pp. 1–240, 2020.

[4] A. Y. Ardiansyah, R. Sarno, and O. Giandi, "Rain detection system for estimate weather level using Mamdani fuzzy inference system," *2018 Int. Conf. Inf. Commun. Technol. ICOIACT 2018, vol. 2018-Janua*, pp. 848–854, 2018, doi: 10.1109/ICOIACT.2018.8350711.

[5] M. H. Gifari, I. Fahmi, A. Thohir, A. Syafei, R. Mardiati, and E. A. Z. Hamidi, "Design and Implementation of Clothesline and Air Dryer Prototype Base on Internet of Things," *Proceeding 2021 7th Int. Conf. Wirel. Telemat. ICWT 2021*, pp. 0–5, 2021, doi: 10.1109/ICWT52862.2021.9678412.

[6] A. Koushal, R. Gupta, F. Jan, K. Kamaldeep, and V. Kumar, "Home Automation System Using ESP32 and Firebase," *PDGC 2022 - 2022 7th Int. Conf. Parallel, Distrib. Grid Comput.*, pp. 228–231, 2022, doi: 10.1109/PDGC56933.2022.10053309

[7] N. Mohd, Z. Hashim, M. Afiq, A. Adnan, and M. D. Sulistiyo, "Automatic Clothesline Puller," pp. 23–26, 2022.

[8] S. Nur Aisyah Abdul Hei, E. Nadzirah Nazri, N. Faqihah Mohamed Rafik, M. Berahim, U. K. Tun Hussein Onn Malaysia, and J. Panchor, "Automatic

Clothesline Retrieval Prototype with Humidity Alert System to Aid Clothesline Drawbacks for Reducing Laundry Worries," *Multidiscip. Appl. Res. Innov.*, vol. 2, no. 1, pp. 401–410, 2021, [Online]. Available: https://doi.org/10.30880/mari.2021.02.01.042

[9] M. I. Ton Mohamad, "Automated Clothesline System," June, 2008.

[10] N. M. H. Siti Dalilah Hashim, Muhamad, "Rain Monitoring And Prediction System Using IoT," 2023, [Online]. Available: doi: https://doi.org/10.1063/5.0110226.

[11] P. Otchere, "Design and Implementation of Automatic Rain Sensing Car Wiper," February, 2020.

[12] [1] Y. A. Ahmad, T. Surya Gunawan, H. Mansor, B. A. Hamida, A. Fikri Hishamudin, and F. Arifin, "On the Evaluation of DHT22 Temperature Sensor for IoT Application," Proc. 8th Int. Conf. Comput. Commun. Eng. ICCCE 2021, pp. 131–134, 2021, doi: 10.1109/ICCCE50029.2021.9467147.

[13] A. A. D. N. Athaya Atsiq Andryan Gunawan, "Automatic Clothing Drying Using Rain Sensors and Ldr Sensors Based on Arduino UNO," vol. 1, p. 2, 2022, [Online]. Available: doi: https://doi.org/10.54482/spectrum.v1i02.174.

[14] M. Nasrulddin, A. Latif, A. Abd Aziz, M. R. Ramdan, and H. Othman, "Design and Development of SMART Automated Clothesline," *Malaysian J. Ind. Technol., vol. 6, no. 1, pp. 1–6,* 2021.

[15] A. Katyal, D. Pathak, and P. Gaur, "An IoT based Real-Time detection and Notification System for an Economical Solar MPPT Model," *ICIERA 2021 - 1st Int. Conf. Ind. Electron. Res. Appl. Proc., pp. 1–6,* 2021, doi: 10.1109/ICIERA53202.2021.9726727.

[16] "ESP32." https://en.wikipedia.org/wiki/ESP32 (accessed Sep. 23, 2020).

[17] "ESP32 Wi-Fi & Bluetooth MCU I Espressif Systems." https://www.espressif.com/en/products/socs/esp32#:~:text=ESP32 can perform as a,SDIO or I2C %2F UART interfaces.

[18] "NODEMCU ESP32," *Cytron Technol. Malaysia*, [Online]. Available: https://my.cytron.io/p-nodemcu-esp32

[19] "Rain Sensor: Working, Pin Configuration and Applications," *ElProCus - Electronic Projects for Engineering Students*, 2019. https://www.elprocus.com/rain-sensor-working-and-its-applications/ (accessed Aug. 04, 2019).

[20] "Buzzer: Working, Types, Circuit, Advantages & Disadvantages," *ElProCus - Electronic Projects for Engineering Students*. https://www.elprocus.com/buzzer-working-applications/ (accessed Jul. 26, 2021).

[21] "5V DC Motor," *HUB360*. https://hub360.com.ng/product/5v-dc-motor/#:~:text=A DC motor is any (accessed May 31, 2023).

[22] S. Prototyping, "Dual H-Bridge Motor Driver L298N." https://www.smart-prototyping.com/L298N-Dual-H-bridge-Motor-Driver-Board

[23] GlossaryTech, "What is Android IDE," *GlossaryTech*. https://glossarytech.com/terms/android/android-ide (accessed Jan. 10, 2018).

[24] Google Developers, "Meet Android Studio," *Android Developers*, 2019. https://developer.android.com/studio/intro

[25] "What is Firebase? All secrets unlocked," 2020. https://blog.back4app.com/firebase/#What_is_Firebase_and_how_it_works (accessed Jul. 25, 2020).

Smart Clothesline

ORIGINALITY REPORT

23%	15%	10%	13%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Siti Dalilah Hashim, Muhamad Hariz Muhamad Adnan, Nor Masharah Husain. "Rain monitoring and prediction system usingIoT", AIP Publishing, 2023 Publication	2%
2	lbeifits.files.wordpress.com Internet Source	1%
3	Submitted to University of San Agustin Student Paper	1%
4	digitalcollection.utem.edu.my Internet Source	1%
5	utpedia.utp.edu.my Internet Source	1%
6	Submitted to University of Sunderland Student Paper	1%
7	www.researchgate.net Internet Source	1%
8	etheses.uinsgd.ac.id Internet Source	1%